AFRL-IF-RS-TR-2002-161
Final Technical Report
July 2002

CLOSED-LOOP HIERARCHICAL CONTROL OF MILITARY AIR OPERATIONS

The Charles Stark Draper Laboratory, Inc.

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J117

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-161 has been reviewed and is approved for publication

APPROVED:

CARL A. DeFRANCO
Project Engineer

FOR THE DIRECTOR:

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
JULY 2002

3. REPORT TYPE AND DATES COVERED
Final Aug 99 – Mar 01

4. TITLE AND SUBTITLE

CLOSED-LOOP HIERARCHICAL CONTROL OF MILITARY AIR OPERATIONS

5. FUNDING NUMBERS
C - F30602-99-C-0206
PE - 63760E
PR - J117
TA - 00
WU - 01

6. AUTHOR(S)

Owen Deutsch, Milton B. Adams, and Janet Lepanto

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Charles Stark Draper Laboratories, Inc.
555 Technology Square
Cambridge MA 02139-3563

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

AFRL/IFSA
526 Brooks Road
Rome NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2002-161

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Carl DeFranco/IFSA/(315) 330-3096

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release: Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

Real-time, closed-loop optimization and control of enterprise –scale dynamic systems remains a challenging problem. Draper's approach combines the theories of decomposition of large-scale optimization problems and distributed control. The structure of the decomposed solution to the optimization problem forms a basis for our controller architecture.

In contrast to ad hoc approaches to decomposing large-scale problems, our approach: (a) results in a distributed system in which situation assessment, problem solving and decision-making across C2 nodes collectively address enterprise-wide objectives; (b) provides significant insight into the nature of the feedback required to "close the loop" around each C2 node in the decomposed problem; and (c) defines the interactions among the C2 nodes to solve the enterprise-wide problem.

Draper developed optimal and heuristic solutions to these problems, implemented a hierarchical control architecture to provide high-rate cyclic and event-based feedback, developed a composite-variable approach to facilitate application of large-scale optimization, and executed experiments to quantify the benefits of higher-rate loop closure and optimization.

14. SUBJECT TERMS
JFACC, hierarchical decomposition, closed-loop planning and control, command and control optimization

15. NUMBER OF PAGES
174

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

Table of Contents

Table of Figures

List of Tables

1 Document Overview

This document is composed of the following remaining sections:

Section 2 *Executive Summary*

This section presents an overview of the report.

Section 3 *Air Strike Operations Problem Statement*

This section describes the problem addressed in this report.

Section 4 *Technical Approach*

This section describes the foundation technology underlying the approach we used to solve the air strike operations problem.

Section 5 *Controller Design and Implementation*

This section describes the controller design and implementation focused on the air strike operations problem.

Section 6 *Modeling and Simulation Implementation*

The section describes the modeling and simulation implementation used to test the performance of the controller implementation.

Section 7 *System Integration – Controller and Simulation Interfaces*

The section describes how the controller and simulation were integrated together to provide an experimental environment.

Section 8 *Experiments*

The section presents the experiments and the results from performing the experiments.

Section 9 *Conclusions*

The section provides a summary with conclusions and suggests specific areas for further research.

Section 10 *References*

The section provides key references.

2 Executive Summary

Real-time, closed-loop optimization and control of enterprise-scale dynamic systems remains a challenging problem. Draper's approach combines the theories of decomposition of large-scale optimization problems and distributed control. The structure of the decomposed solution to the optimization problem forms a basis for our controller architecture. In contrast to ad hoc approaches to decomposing large-scale problems, our approach:

a) results in a distributed system in which situation assessment, problem solving and decision-making across C^2 nodes collectively address enterprise-wide objectives

b) provides significant insight into the nature of the feedback required to "close the loop" around each of the C^2 nodes in the decomposed problem, and

c) defines the dynamics of the interactions among the C^2nodes in solving the enterprise-wide problem, including the objectives passed from higher-level nodes to lower-level nodes and the feedback/status passed from lower-levels to higher-levels.

For air operations strike-planning, the levels of the decomposed problem are: *Target and Aircraft Allocation* allocating targets (i.e., tasks) to be pursued and aircraft resources to be used in each subproblem, *Mission Generation* assigning and scheduling specific aircraft package and weapon resources for specific targets, and *Routing* generating optimal routes for attrition, time, and fuel from bases through tankers and assembly point to and from targets. Draper developed optimal and heuristic solutions to solve these problems, implemented a hierarchical control architecture that provided both high-rate cyclic and event-based feedback for this multi-level optimization problem, developed a composite-variable approach that represents an enormous advance in the ability to apply large-scale optimization to these problems, and executed an extensive series of experiments to quantify the benefits of higher-rate loop closure and optimization. The experiments address performance, robustness to uncertainty, and stability and represent one of the first instances where the benefits of higher-rate loop closure have been quantified for military command and control.

2.1 Problem Statement

Military air operations entail command and control of diverse forces distributed over potentially large geographic areas. This geographic distribution, coupled with the need for short decision cycle times, requires an agile, distributed, and collaborative command and control (C^2) system capable of dynamically tasking: (1) aircraft to strike targets; (2) supporting logistics; and (3) sensing and electronic ISR assets. Our focus here is on the following *coupled* problems related to air strike including: (1) ***assigning strike resources*** (i.e., aircraft and weapons) ***to targets***; (2) ***scheduling the application of those resources over time***; and (3) ***determining fuel-feasible, minimum risk routes*** for those resources.

In the air strike operations plan, aircraft are tasked to work together in groups called strike packages where each aircraft in a strike package performs specialized functions that contribute to the overall effectiveness of the package. For instance, a strike package may consist of bombers escorted by jammers. The jammers in the strike package reduce the effectiveness of enemy air defense radar that could detect the bombers en route to their targets. The components of a strike package that are appropriate for a given mission objective depend on the targets to be struck, the type and level of enemy resistance expected en route and at the targets, and the competition for strike resources from other mission objectives.

Our system should enable commanders at the wing operations level and at the strike package level to plan and execute integrated air operations in shorter time frames and with greater effectiveness than is now possible. The system architecture uses information as it becomes available and ensures that local command decisions contribute toward optimization of enterprise-wide strike objectives. The system produces an executable set of integrated tasks for strike aircraft assets that automates and optimizes the use of resources, responds quickly to new information, supports distributed decision-making, and enables Commanders to generate, execute, and adapt strike operations well within the decision cycle of potential adversaries.

Repetitive and tedious, low-level interactions will be eliminated by automation, although operators, through an appropriate interface (not developed under the current effort), will be able to modify plan outputs or initialize plan inputs with low-level specifications as desired. Our system will provide allocation, assignment, and scheduling for formulating executable air asset tasking, including routing for threat avoidance, strike package selection, and weaponeering selection.

Air operations plans are driven by Commander's Intent, where Commander's Intent reflects:

- ***Time***: importance of campaign phase (time phase importance or target time criticality).
- ***Geography***: importance of region.
- ***Target Class***: importance of target class or grouping of targets.
- ***Risk***: the importance of achieving objectives vs loss of resources (including human).

The scope of the air operations planning problem that we address includes:

- Deciding which targets to hit and when.
- Deciding which assets to use to deliver weapons.
- Deciding routes to follow, refueling, and assembly points.
- Assigning wild weasel and jammer escorts.
- Respecting the laws of physics, logistics, and human performance.
- Accounting for risk in decisions.

The objective is to accomplish damage to specified target systems in a manner that:

- Minimizes operational costs including attrition.
- Satisfies operational requirements such as timeliness of action.
- Allocates capacitated resources between targets in a manner reflecting Commander's Intent.

Major assumptions include:

- Campaign objectives express Commander's Intent.
- Estimates of the state of the air strike resources and their operational environment are provided to the controller at regular intervals.

2.2 Technical Approach

Fundamental to our approach is explicit modeling of Commander’s Intent in the form of tables. The numerical values from JFACC Commander’s Intent tables are applied to a functional transformation that yields a relative valuation for each target at each time as a function of the overall target damage state. This is central to the planning algorithms that task and schedule resources to maximize the aggregate objective function as well as to execution-monitoring algorithms that control dynamic replanning. It is anticipated that the Air Operations staff would customize the data in these tables based on the JFACC Commander's guidance and stated numerical objectives. The Commander’s Intent tables include the following tables, where all entries are indexed by geographic region and target functional category:

- A table that expresses relative target valuation.
- A table that expresses probability of damage objectives.
- A table that expresses target time-perishability or criticality time constants.

- A table that expresses target time-reconstitution time constants.
- A table that indicates the degree of target coupling.

Because of the size and complexity of military air operations, it is intractable to develop a single controller that is capable of producing detailed plans for every entity involved in air operations. It is also futile to generate detailed plans for problems over extended durations because of the high probability of future events deviating from current expectations. We employed an approach to decomposing and executing the problem that combines the theories of decomposition of large-scale optimization problems and distributed control. Unlike ad hoc approaches to decomposing large-scale operational problems, our approach results in a distributed system for which the problem-solving and decision-making within each distributed controller addresses enterprise-wide objectives. This approach to decomposition both provides significant insight into the nature of the feedback required to close the loop around each of the controllers within the decomposed problem, and defines the dynamics of the interactions among the controllers in solving the enterprise-wide problem, including the objectives passed from superior nodes to subordinate nodes and the feedback/status passed from subordinates to superiors. Each controller generates plans that are coordinated by a supervisory controller in order to ensure that the combined solution adheres to the constraints and objectives provided to the overall planning system.

Each command and control node within the controller architecture is functionally decomposed, as represented in Figure 2.2-1. Sensing the plant, i.e., the "system to be controlled" and the environment provide feedback. The "system" may be physical entities within the plant that are being controlled, or it may represent an aggregation of lower-level problem solving nodes along with the entities they control. A closed-loop, hierarchical decomposition is a recursive implementation of the node illustrated in Figure 2.2-1, where the "system-to-be-controlled" is one or more subordinate-level processes that are "controlled" or coordinated by an upper Master level, as shown in Figure 2.2-2.

The feedback should contain the information required to evaluate progress toward the solution to the subproblem being solved. Since the solutions generated will span a finite time horizon, models will be required to predict future states and status based on the planned course of action and estimates of the current state.

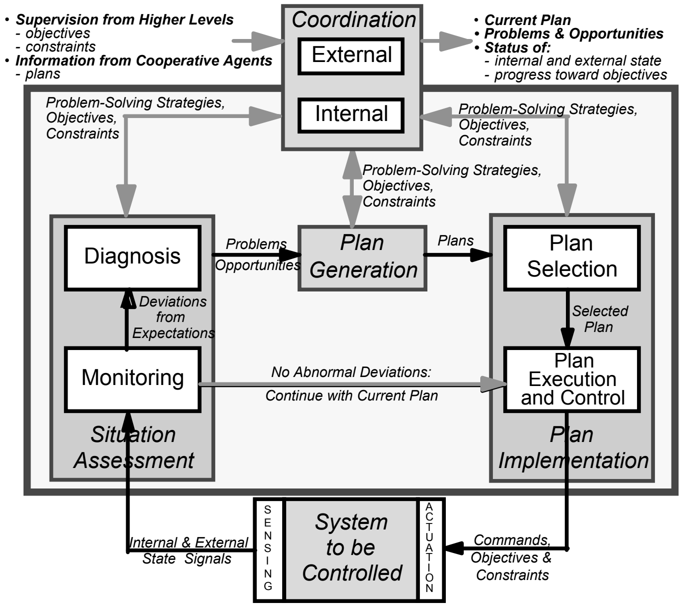

• Supervision from Higher Levels
- objectives
- constraints
• Information from Cooperative Agents
- plans
Coordination
External
Internal
• Current Plan
• Problems & Opportunities
• Status of:
- internal and external state
- progress toward objectives
Problem-Solving Strategies, Objectives, Constraints
Problem-Solving Strategies, Objectives, Constraints
Problem-Solving Strategies, Objectives, Constraints
Diagnosis
Problems
Opportunities
Plan Generation
Plans
Plan Selection
Deviations from Expectations
Selected Plan
Monitoring
No Abnormal Deviations:
Continue with Current Plan
Plan Execution and Control
Situation Assessment
Plan Implementation
SENSING
System to be Controlled
ACTUATION
Internal & External State Signals
Commands, Objectives & Constraints

Figure 2.2-1. Functional Decomposition of a Command and Control Node.

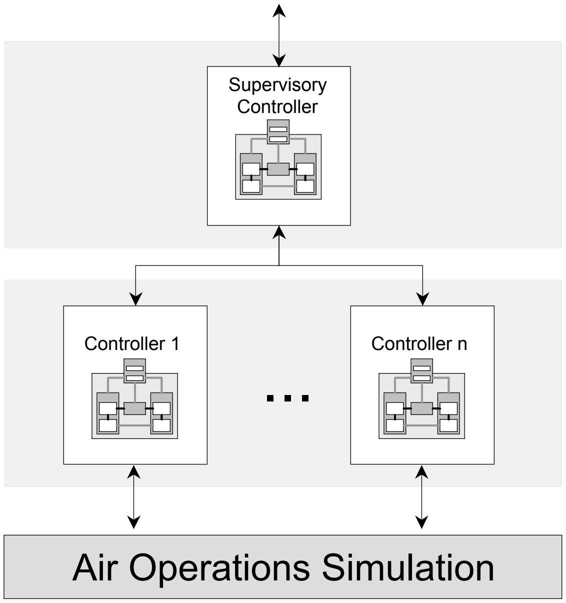


Figure 2.2-2. Hierarchical View: Aggregated Plant for Master Supervisory Level.

The closed-loop ***optimal control*** for each of the distributed elements is formulated as a receding horizon optimal control problem with the following attributes:

- ***Optimization:*** a time-varying "set point" control (e.g., a plan) and associated expected time history of system state values that would evolve if the plan were pursued are determined over a finite time horizon to optimize an objective function representing the desired system performance (in our case, the Commander's Intent).
- ***Control***: a high rate "perturbation controller" will augment the set point commands to stabilize the operation of the system and to ensure that the state of the system tracks the trajectory associated with the set point in the presence of disturbances.

Our approach achieves two objectives:

1. Provides a structure/framework for solving the air operations problem that resolves both resource conflicts across lower levels, as well as allocates objectives to the lower levels.

   This is done in a way that results in a degree of autonomy on the part of the lower levels in solving their decoupled problems. There are human-in-the-loop considerations on how one maps the decomposed problem onto the human organizational elements that must be ultimately responsible for planning and execution. Our approach defines the type of negotiation (e.g., iteration) and associated information exchanges among the levels required to arrive at a good, overall solution.

2. Uses the data/information exchanges among problem-solving elements prescribed by the decomposition to form the basis of the real-time feedback when we "close-the-loop" on the decision-making.

   That is, in accommodating our inability to exactly model/formulate the problem due to the myriad of uncertainties and unknowns that prevail in a real warfighting situation, the sensitivity to those uncertainties is reduced using feedback (the purpose of closing the loop in even the simplest of control systems). Thus, the solution will evolve over time in response to the sensed state as well as new objectives provided by command levels.

2.3 Air Operations Models

Development and testing of the Air Operations Controller utilized a numerical simulation with models for aircraft, air bases, ground-based air defense threats, weapons, and targets. Information elements include the state of objects, including locations, fuel status, damage status, as well as commands that determine intended future actions. The simulated execution of commands is conditioned by constraints such as proximity, fuel state, state of ordnance stores, stochastic factors, etc. For example, a commanded weapons release is not executed if the simulated aircraft has not arrived within proximity of the prescribed release point or is not carrying the prescribed weapon type. Similarly, commanded aircraft recovery will not occur if not within proximity of an operating base, commanded refueling will not occur unless within proximity of the prescribed tanker location, and aircraft that run out of fuel because of planning deficiencies will be reported as lost.

Potential human decision elements include the decisions and actions of JFACC planners, pilots, and air defense system operators. The actions of JFACC planners are abstracted out across the interface that defines the experiment and scenario. In other words, the parameters and controls on the JFACC Air Operations Controller are initialized, and any changes during an experiment are scripted. The JFACC Air Operations Controller operates in a fully autonomous mode during the experiment, with perturbations representing human interactions either scripted or represented by stochastic models using pseudorandom numbers. This is necessary to allow the replication of experimental conditions and to provide other aspects of a systematic experimentation environment.

Pilots are assumed to properly execute commanded mission tasks and the enforcement of physical constraints and proximity tests are assumed to parallel the actions of human pilots. Simulation resolution does not encompass detailed tactics or other detailed actions mediated by human judgement. Pilot performance in target acquisition is modeled stochastically. There is no simulation of pilot-based replanning or reaction to contingencies. The decisions of air defense system operators is folded into the overall performance model of air defense threats, with a stochastic model for the likelihood of engagement given a physical condition of engageability.

- The overall technical approach to Air Operations Simulation is to apply the lowest fidelity and least complex model that will represent desired behaviors and capture desired interactions with the JFACC Air Operations Controller. In this vein, aircraft dynamics are represented by great circle, constant speed propagations between commanded waypoints and details such as take-off roll and acceleration to cruise speed are not explicitly modeled.

The principal models are:

- Time duration for activities, such as ground preparation, aerial refueling, target acquisition and weapon release, aircraft recovery.
- Great circle constant speed propagation between commanded waypoints, with banked-turn loitering on waypoint arrival prior to the specified time of arrival.
- Fuel use and remaining unrefueled endurance that is linear with time.
- Target fractional damage that is deterministic according to weaponeering specifications.
- Stochastic air defense threat interactions.

Each aircraft is separately tasked, with mission package coordination handled solely by the Air Operations Controller. Mission packages often but not always originate from the same base, with provision for dynamic retasking and dynamic package formation at assembly points. There is no "Commander logic" or rule-based system for aircraft autonomously determining mission aborts and initiating return to base behavior. Absent any error in the mission tasking, there is no contingency in the Air Operations Simulation where this would be required. Threat interactions result in either no change or a shoot-down of mission aircraft. Aerial refueling is explicitly tasked. The Air Operations Controller includes a significant safety margin of flight endurance to mitigate errors in fuel use models.

2.4 Controller Design and Implementation

The overall objective of the air strike operations planning problem to be solved by our controller is to maximize the target value achieved over a specified planning horizon subject to the constraints of the air operations environment and the available resources. This section gives the general mathematical formulation of the problem being solved.

Objective: select from among the available targets (**T**), the available aircraft packages (**P**), the feasible routes (**R**) for the aircraft composing those packages, and the feasible weapon configurations (**W**) for those aircraft that maximize the total expected return as defined by:

$$\mathop{Max}\limits_{\substack{P \in \mathbf{P} \\ R \in \mathbf{R} \\ W \in \mathbf{W} \\ T \in \mathbf{T}}} \sum_{tar_i \in T} \Big[E(v(tar_i)) - E(\cos t(tar_i)) \Big]$$

where

P = the set of selected aircraft packages from all available packages **P**

R = the set of routes chosen for the aircraft in P from the fuel-feasible routes **R**

W = the weapon configurations (weaponeering) chosen for each of the aircraft in P from among the feasible configurations **W**

T = the set of targets selected from the available set **T**

tar_i = the i^{th} target in the target set T

$v(tar_i)$ = the value to the campaign accrued for target tar_i when the desired level of damage is achieved

$E(v(tar_i))$ = the expected value of target tar_i where the expectation is taken over the uncertainty in reaching, finding, and successfully damaging the target to the specified level. This expectation is a function of the air defense threat, the selected package (e.g., level of escort), the selected routes for the aircraft within the package, the selected weaponeering for the package, and the time sensitivity (e.g., mobility) of the target, and the time that the routes will be executed

$E(cost(tar_i))$ = is the expected cost of prosecuting target tar_i where the expectation is taken over the attrition cost of attacking the target and is a function of the chosen routes, aircraft package (e.g., escort level), and air defense threat laydown. Also included in the cost are aircraft flight hour costs and fuel cost, both of which are modeled as deterministic

Constraints: The constraints for the problem relate to the availability of aircraft and weapons, performance of aircraft and weapons, and the locations of tankers available for refueling:

- Aircraft and weapons resource supply.
  - Aircraft supply constrains the aircraft packages that can be assigned.
  - Weapon supply constrains the weaponeering that can be assigned.
- Aircraft and weapon performance.
  - Aircraft performance constrains the routes aircraft can fly, including the speed at which the aircraft can fly and distance they can travel without refueling.
  - Aircraft performance constrains the weapons aircraft can carry.
  - Aircraft performance constrains the effectiveness of using escorts.
  - Weapon performance constrains the effectiveness of weapons against individual targets.
- Tanker Locations.
  - We assume tanker locations are prespecified, which constrains routing and aircraft package to target assignments.

We decomposed the formulation using an *interaction prediction* approach. An upper-level C^2 node assigns targets to bases with each base responsible for determining the strike packages and their associated routes to assign to each target. Such a decomposition is justified because the detailed planning required of a base to execute a strike is strongly influenced by its local resource constraints, and the information resulting from this detailed planning is not easily represented in data that can be readily communicated to or used by the upper level.

The decomposition comprises a master problem and a set of subproblems. For our decomposition, each subproblem is associated with a base, and the master problem is associated with the upper-level C^2 element. The base problems cannot be solved independently without coordination from the upper level because without coordination, multiple bases would likely strike the same target in the same wave, violating a constraint that each target be struck by no more than one strike package. The upper-level coordination addresses this issue by assigning nonoverlapping sets of targets to the bases. The controller hierarchical decomposition is illustrated in Figure 2.4-1.

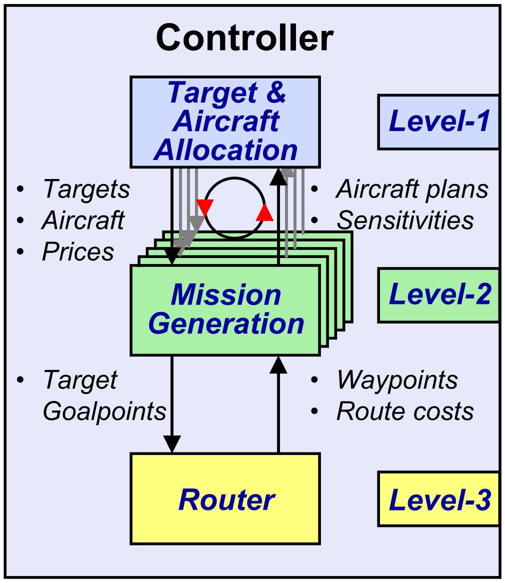


Figure 2.4-1. Controller Hierarchical Decomposition

A hierarchical architecture of subcontrollers that mirrors the decomposed JFACC problem has been developed and applied to our family of JFACC scenarios. The architecture is composed of basic building blocks (subcontrollers), each incorporating the functions of "observe-orient-decide-act," and which are termed in this report as "monitor-diagnose-plan-execute." At the higher levels, planning time frames are long, objectives are broad, and decisions are aggregated. At lower levels, the reverse is true.

This controller structure was chosen to take advantage of its good properties:

1. ***Problem Tractability:*** Decomposing the problem into stratified levels over space, as well as time, is often the only way a problem can be solved.

2. ***Conformance to Human Organizations:*** Quite often, new decision-making techniques need to operate within an existing human organization.

3. ***Reduction in Computation Time*:** Hierarchical decomposition should always reduce the time required to solve a problem from that needed to solve a single, monolithic formulation of the problem. Even when

single formulations are solvable, time constraints on execution may argue for the time savings that can be obtained through decomposition.

4. ***Scalability:*** Hierarchical structures are extraordinarily scalable, as evident from their wide application in society, from small companies to large military organizations.

The main penalty paid for employing a hierarchical approach is that design and implementation of a system of controllers can be more involved than that of a single controller node. The subcontrollers in the hierarchy must be designed to be highly cooperative and interactive, accepting compatible objectives and constraints from above, and listening for problems and opportunities from below. Also, the orchestration of ongoing activity within the hierarchy is more demanding when a number of subelements need to work together. But the implementation of this orchestration is a one-time cost that can be spread over multiple applications once it is developed.

Both heuristic and optimization algorithms were developed to solve the master problem and subproblems illustrated in Figure 2.4-1.

The ***heuristic Target and Aircraft Allocation (Level-1) planner*** allocates targets to the Mission Generation (Level 2) subproblems to encourage the generation of plans that prosecute high-valued targets in a timely manner while maintaining workload balance among subproblems. A target list is created and ranked in order of decreasing target value. The Level-1 planner allocates each target on the target list to the nearest base, starting with the highest-valued target, thereby allowing higher-valued targets to be prosecuted sooner. Workload balance is maintained by ensuring that the number of targets allocated to each base is proportional to that base's workload capacity. The workload capacity is measured as a function of that base's weapon delivery capacity.

In addition, the Level-1 planner is capable of negotiating with the lower-level planner to improve on the target allocation. Negotiation is an iterative process in which the lower levels pass sensitivity information and completed plans to the master level. The master level processes the sensitivity information to determine a new allocation of targets and resources to subproblems. These modified subproblems are returned to the lower levels for planning, resulting in plan improvements.

The ***heuristic Mission Generation (Level-2) planner*** prioritizes targets, selects the weapons and strike package configuration to be applied to each target, assigns aircraft to targets, and schedules sorties. This planner attempts to maximize target value specified by region and functional category as established via the Commander's Priority Input Matrix. At the same time it attempts to minimize a total cost function that accounts for attrition risk, operational costs per flight hour, weapon utilization, and reduction in combat effectiveness when pilots are re-tasked during flight.

The heuristic implementation of the Mission Generation planner serves two functions. First, it provides a baseline against which to compare the performance of the optimal integer programming solution. Second, it provides optimal or near-optimal candidate missions for conditions that the optimal planner specifies. In this second function, the heuristic planner forms the lower level of a decomposition that greatly speeds convergence of the optimal planner.

The heuristic algorithm sequentially assigns aircraft to targets, optimizing the incremental contribution to an objective function for each aircraft mission that the planner generates. This "greedy" approach is sub-optimal, but it generates reasonable plans quickly. The heuristic planner uses expected target value divided by expected total mission cost for each mission (combination of a target and the strike package assigned to that target).

The Level-3 planner is a ***Strategic Router*** that supports the Level 2 planner (Mission Generation planner) by providing the cost of constrained minimum risk routes for specified aircraft-target pairs.

The route-planning problem is:

Given a set of

1. **Mission parameters** including:
    - Start location (base or en route) and return base.
    - Required ingress and/or egress assembly points.
    - Target location.
    - Set of all tanker locations.
2. **Aircraft parameters** including:
    - Fuel endurance.
    - Pilot endurance.
3. A **Threat model** including:
    - Threat density.
    - Detection range.
    - Likelihood of engaging.
    - Possibility of attrition in an engagement (P_k)
4. **Escort level** representing one of:
    - No escort.
    - Wild weasels.
    - Weasels plus escort jammers.

Determine a strategic-level route (10- to 30-km grid spacing for waypoints) that:

- Is feasible with respect to aircraft fuel endurance.
- Minimizes risk from threat engagement.
- Allows for a maximum of two refueling activities on each segment.
- Adheres to specified assembly points.

Given the underlying threat model, the router was implemented using an A* search to produce fuel-feasible minimum attrition cost routings according to a mission structure that includes:

- a vertex corresponding to takeoff from a base or an arbitrary enroute start location
- a possible vertex for an ingress assembly point
- a vertex at the target weapon release location
- a vertex at a possible egress assembly point
- a vertex at the specified return base.

In addition to generating the min-cost route and the planned arrival times at all waypoints, the router reports the estimated attrition risk along the determined path. This result is used in the Mission Generation heuristic to enforce the risk management constraint. Candidate missions that exceed a risk level that is specified as a function of aircraft type are rejected by the heuristic.

2.5 Modeling and Simulation Implementation

As illustrated in Figure 2.2-2, the JFACC Air Operations Controller and the associated plant can be viewed as a hierarchy of processes with some of these processes distributed over different spatial locations. This could be implemented as a hierarchy of simulated processes with associated controllers and the implementation could also be distributed. The current implementation, however, imposes the simplification of a single, global plant simulation that simulates the execution of the entire enterprise as a single computational process. Similarly, the current controller implementation, although inherently hierarchical in function, is implemented as a single computational process, separate however from the simulation process. The reduction in software complexity occasioned by this implementation decision greatly facilitates the execution of experiments as well as the initial development of all the simulation and controller software.

The state of the plant includes the following own force elements:

- Air bases.
- Tanker locations.
- Attack and strike package escort aircraft.
- Weapons.

The state of the plant relating to hostile forces includes:

- Targets.
- Air defense threats.

Finally, the environment is a spherical Earth geography that determines transit times given aircraft speeds and routes as specified by sets of waypoints. The air bases have a state that is either operational and available to launch missions or unavailable except for emergency recovery of aircraft. Refueling tankers are not explicitly modeled except by a location to which aircraft must fly to take on fuel. Tanker missions are not explicitly tasked, and tanker loadouts are not modeled. The tanker requirements that result from the Air Operations plan are recorded.

The state of strike aircraft includes the maintenance-conditioned availability for mission tasking, the location, the fuel state, and the current weapons loadout. This is initialized to one of a set of standard conventional loads and then decremented on weapon release. Weapons are not modeled explicitly, but serve only to count capacitated resources.

Target state is modeled by a location, a current damage level, and a weaponeering specification that indicates how many of which weapon types are required to produce specified damage levels.

The air defense threat state is modeled only as a spatially-varying density of threat systems with associated engagement and lethality parameters.

The controller is assumed to have perfect information on own forces state, but may have incomplete or erroneous information on adversary targets and threats. Specifically, targets are disclosed at scripted times to the controller and may be available for attack only within a specified time frame. In the highly aggregated model, it is assumed that target acquisition opportunities vanish with a first-order exponential model from the time of target disclosure. Targets must be acquired before own force strike aircraft can release weapons and cause damage on those targets. Additionally, target weaponeering specifications sent to the controller may contain errors with respect to the simulated truth model. Finally, the estimated air defense threat density sent to the controller may contain discrepancies with respect to the simulated truth model. The effects of these discrepancies on controller performance are assessed in a specific set of experiments.

In summary, the classes of plant disturbances to be considered include:

- Unanticipated changes in own resources due to unexpected rates of attrition and unanticipated temporary base closures.
- Unanticipated level of effectiveness of own weapons employed.
- Unanticipated level of air defense threat effectiveness.
- Unanticipated changes in adversary activities reflected in changing target locations and values as reflected in the Commander’s Intent.

Thus, disturbances are principally due to errors in modeling the evolution of plant state components; to uncertainty in exogenous inputs to the plant, such as adversary command and control directives; and to changes to own force command and control directives, such as changes in Commander's Intent weightings across phase boundaries.

The principal elements of our plant model are expressed by the primary aircraft activity functions:

- Ready Aircraft in a specified mission configuration and take-off on activity completion.
- Go To Location specified by a great circle route to the next waypoint, depleting remaining fuel endurance in transit.
- Refuel Aircraft if in proximity to a specified tanker location.
- Release Weapons specified for a particular target if in proximity to specified release location.
- Recover Aircraft if in proximity to an air base location.

Each of these activities has a logistical consequence, a specified activity duration, a specified earliest starting time and specified latest completion time, and a location proximity relation that must be satisfied to either initiate or complete the activity. Aircraft that arrive at a waypoint before the specified earliest start for proceeding to the next waypoint or activity execute a banked-turn loiter at that waypoint.

The air defense threat interaction model is evaluated for all aircraft in the air. One component of the model is the determination of physical proximity of defense suppression escort aircraft. A second component stochastically samples for attrition outcome from probabilities evaluated from the threat density, the threat parameters, the aircraft track, and the aircraft escort status. The details of this model will be discussed shortly.

The simulation implements controller commands, updating aircraft and target states according to the execution of planned activities. In the process of executing the missions, the simulation monitors fuel, pilot endurance, weapons, activity duration, proximity to base, tankers, and determines target damage status. Figure 2.5-1 is a snapshot of one of our simulation displays.

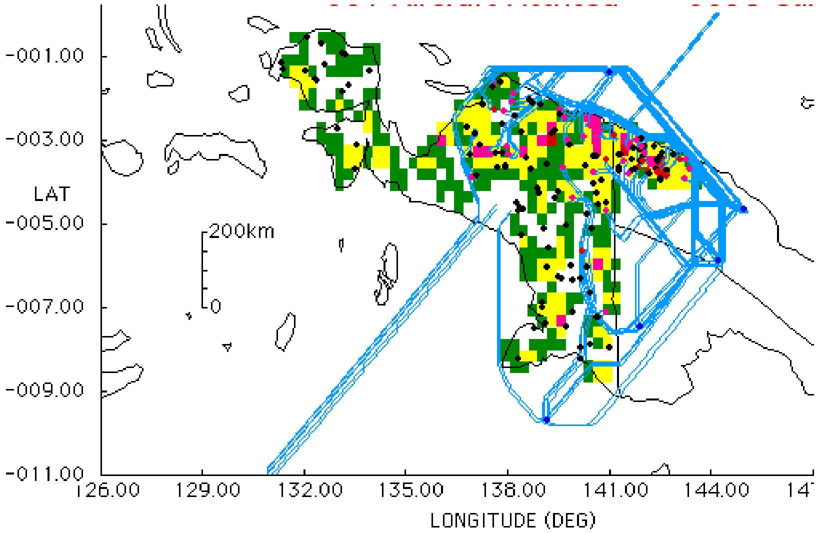


Figure 2.5-1. Simulation Display of Aircraft Routes.

The original scenario storyline included an invasion by West Cyberland, a former Soviet-bloc client with a large military and inventory of weapons, of East Cyberland, a U.S. ally with a modest military and significant production of oil in the North central portion of the country. The invasion forces meet with little resistant and quickly establish an invasion salient headed toward the East Cyberland oil fields.

The top-level U.S. objectives are to halt and roll back the invasion, preventing seizure of or damage to East Cyberland oil fields, if possible. The U.S. has a couple of aircraft carrier battle groups (CVBGs) at distances requiring 2 and 5 days steaming to the theater of operations. It is assumed that tactical strike aircraft can be staged into several East Cyberland in-country bases that are not close to being overrun to support high-tempo operations against the invasion and against supporting infrastructure in West Cyberland. Additionally, bases in Guam and Darwin, Australia, at about 2 h flying time from the theater, are assumed available for staging tanker, long-range bomber, ISR, and other supporting missions.

2.6 System Integration

As shown in Figure 2.6-1, the JFACC Air Operations Simulation and the JFACC Air Operations Controller interface to each other through a set of shared interface files, and there are also several other files that are integral to the definition and interpretation of experiments. The sequence of operation is as follows:

- The user prepares the Initial Data or Scenario file for reading once by the simulation.
- The simulation generates a state file to be sent to the controller.
- The controller generates a command file that is read by the simulation.
- The simulation executes the command file, generates a number of output files for analysis, debugging and replay, and repeats the cycle of sending the state file to the controller.

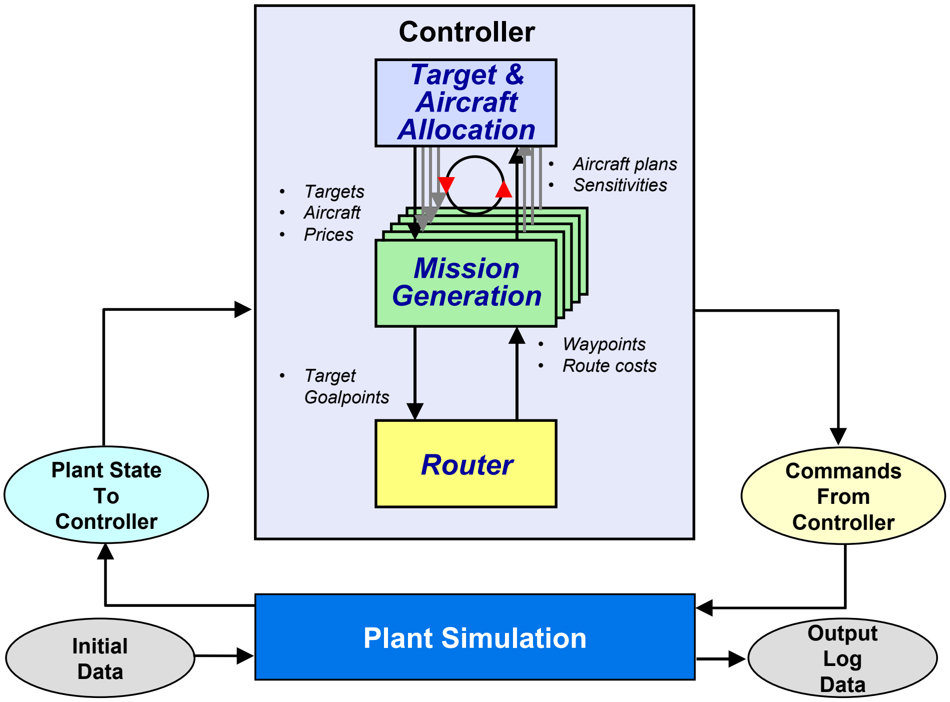


Figure 2.6-1. Controller –Simulation Architecture.

There are additional files required by the simulation that do not change from experiment to experiment including:

- World vector shoreline and political boundary files for map visualization.
- File of random number seed values.
- File of waypoints and specifications for externally scripted objects.

The externally scripted objects file is not changed between experiments since this capability is not currently used in the experiments that we have defined.

Everything needed to define a scenario is included in the scenario file. The data items will be described in detail shortly, and will be seen to be very similar although not identical to the data items on the state file.

Finally, to facilitate the unattended execution of a number of long-running experiments, a "Production Run" file is defined that specifies the filenames of a number of scenarios along with perturbation and Monte Carlo parameters.

2.7 Experiments and Results

A set of experiments has been designed to investigate the air operations effectiveness that results from the application of our distributed control architectures. Specifically, these experiments investigate:

- The improvements gained and sensitivity to the rate of loop closure.
- The robustness of our closed-loop architecture to uncertainties in the models employed for blue aircraft attrition and weapons effectiveness.
- The agility of the close-loop system response to command-level designation of new targets and/or changes in priority/valuation of existing targets.

Experimental results are obtained by applying our hierarchical controller to a simulation of a military air operation scenario, wherein only the salient characteristics of military air operations are modeled. The experiment scenario has been chosen to provide an operational setting wherein we can credibly illustrate that feedback and shorter control cycle times yield improved performance and robustness to modeling uncertainties and a changing operational environment. Randomization via a small number of Monte Carlo trials has been employed to assess the performance and robustness of our closed-loop controller design.

The state of the plant has been limited to blue (friendly) and red (adversary) forces, with the principal *blue* state elements being air and supporting resources, including bases and weapons stores; and the principal *red* state elements being air defense, targets, and supporting components. The intent is to control the evolution of the blue state and influence the evolution of the red state.

The classes of plant disturbances include:

- Unanticipated changes in blue resources due to unexpected rates of attrition.
- Unanticipated discrepancy in effectiveness of weapons employed.
- Unanticipated changes in red activities reflected in changing target locations and values.

The primary metrics employed for the evaluation of experiments are those ***associated with the accomplishment of the Commander's Intent***. Those metrics include the aggregate value of target destruction by: (1) target category, (2) operational geographic region, and (3) campaign phase (time), as well as (d) time to achieve levels of fractional destruction along these same dimensions. On the cost side, the attrition of aircraft and the cost of utilization of munitions and mission support resources is logged and included as an element of the evaluation for each experiment. The results presented here focus on the aggregate target damage metric.

In addition to cost and plan value, we also evaluate the performance of our closed-loop controller in the context of "plan stability." Here, plan stability is a measure of how plans change each time the loop is closed and new plans are developed. From a human factors perspective, it is unacceptable to have frequent, significant changes in strike plans for individual aircraft, especially when they are already en route to a target.

2.7.1 Experiments

The following two tables summarize performed experiments.

Table 2.7-1 summarizes the set of cyclic loop closure experiments that were performed. The baseline scenario is the Cyberland Scenario with 313 targets disclosed over five days of campaign as described in Section 6.

Table2.7-1. Cyclic Controller Experiment Summary.

Closed Loop Cyclic Experiment	***Controller Configuration***	***Scenario Variation***	***Experimental Objective***	***Results***
Problem Decomposition	Undecomposed and Decomposed	Baseline	Determine the computational benefits of decomposing the problem	Reduce computational time from decomposition quantified
Loop Closure Rate	4 hour and 24 hour loop closure intervals	Baseline	Determine the performance benefits of higher rates of feedback	Improved performance from higher rate feedback quantified – significant shortening of time to achieve objectives
Sensitivity to Modeling Errors				
- *Weapons Effectiveness*	4 hour loop closure interval w/ Effectiveness of Weapons underestimated in the controller's model	Weapons Effectiveness reduced by a factor of 2 - 5	Determine the robustness of the closed-loop controller to errors in weapons effectiveness modeling	Higher rate feedback enables more rapid reattack of high-value targets based on BDA, improving performance robustness to weaponeering errors
- *Air Defense Effectiveness*	4 hour loop closure interval w/ Effectiveness of Enemy Air Defense underestimated in the controller's model	Lethality of Enemy Air Defense increased by a factor of 2	Determine the robustness of the closed-loop controller to errors in enemy air defense effectiveness modeling	Higher rate feedback improves robustness to errors in attrition modeling, resulting in improved target damage and reduced attrition.
Sensitivity to Unexpected Scenario Changes				
- *Commander's Intent*	4 hour loop closure interval	Unexpected change in target values by category and region due to change in Commander's Intent	Determine agility of the closed-loop controller in responding to target value changes	Higher rate feedback responds more rapidly to changed campaign objectives.
- *Base Closures*	4 hour loop closure interval	Unexpected closure of an air base due to poor weather conditions	Determine agility of closed-loop controller to reconfigure air operations plans in the face of base closure	Higher rate feedback enables rapid reconfiguration of plans to large scale perturbations such as base closures
All UCAV Fleet	4 hour loop closure with reduced sensitivity to risk and reduced cost of retasking aircraft in flight	All aircraft unmanned	Determine the improvement in performance and increased attrition when all aircraft are unmanned	Higher rate feedback enabled more effective use of aircraft with low risk aversion and low replanning costs
Time Sensitive Targets	4 and 24 hour loop closure intervals	Variations include: all TSTs, no TSTs, all TSTs and no Air Defense Threat	Determine ability of the closed loop controller to plan and execute missions for various levels of TSTs	Quantified effect of higher rate feedback to improved performance on time sensitive targets

Table 2.7-2. Event-Based Controller Experiment Summary

Event-Based Controller Experiment	*Controller Configuration*	*Scenario Variation*	*Experimental Objective*	*Results*
Comparison to Cycle	Event Based/4 hour cyclic	Baseline	Determine performance benefits of the controller reacting (replanning) as events occur	Demonstrated improved ability to prosecute time critical targets
Increased Air Defense Threat	Event Based with negotiation /4 hour cyclic w/o negotiation (both controllers correctly model increased threat)	Air defense lethality increased by a factor of 5	Determine improvement due to more timely response to attritted aircraft in a package	Demonstrated improved ability to achieve target damage and manage attrition
All UCAV Fleet	Event Based/4 hour cyclic Both with reduced sensitivity to risk and reduced cost of retasking aircraft in flight	All aircraft unmanned	Determine ability of the closed loop controller to plan and execute missions for various levels of TSTs	Benefits of employing non-risk averse aircraft for time sensitive targets was demonstrated for event-based as well as cyclic planners

2.8 Conclusions

The work reported here is one of the first instances where the benefits of higher rate loop closure have been quantified for a complex enterprise command and control application such as coordinating air attack operations, spanning the air operations enterprise from JFACC level to the strike package level. Our experimental results show that the benefits are substantial, and that they accrue even in the face of the types of model discrepancies that are to be expected in such applications. We should note that the results reported here assume perfect state estimation and feedback for own forces as well as for BDA.

In summary we found that:

- **Effective closed-loop solutions to realistic large-scale air operations planning problems can be largely automated.**
  - Automation allows decreasing controller cycle time from 24 h to 4 h.
  - Automation reduces the time to achieve Commander's Intent.
  - Closed-loop automation reduces sensitivity to unanticipated events.
- **Problem decomposition results in substantial reduction in computation time with little loss in performance.**
- **A controller can be designed to properly incorporate:**
  - Commander's Intent and user-definable levels of risk aversion.
  - Plan stability across successive planning cycles.
- **Optimal formulations for this class of problems are difficult to tractably solve, but can be used to help develop and evaluate heuristic algorithms that perform well.**

3 Air Strike Operations Problem Statement

Military air operations entail command and control of diverse forces distributed over potentially large geographic areas. This geographic distribution, coupled with the need for short decision cycle times, requires an agile, distributed, and collaborative command and control (C^2) system capable of dynamically tasking: (1) aircraft to strike targets; (2) supporting logistics; and (3) sensing and electronic ISR assets. Our focus here is on the following *coupled* problems related to air strike including: (1) assigning strike resources (i.e., aircraft and weapons) to targets; (2) scheduling the application of those resources over time; and (3) determining fuel-optimal, minimum risk routes for those resources.

In the air strike operations plan, aircraft are tasked to work together in groups called strike packages where each aircraft in a strike package performs specialized functions that contribute to the overall effectiveness of the package. For instance, a strike package may consist of bombers escorted by jammers. The jammers in the strike package reduce the effectiveness of enemy air defense radar that could detect the bombers en route to their targets. The components of a strike package that are appropriate for a given mission objective depend on the targets to be struck, the type and level of enemy resistance expected en route and at the targets, and the competition for strike resources from other mission objectives.

The objective of the strike-planning problem is to assign resources to targets in a way that maximizes the expected accomplishment of mission objectives defined by the Commander's Intent as described below and later in more detail in Section 4.1. We assume that no more than one strike package strikes a target over a specified planning horizon, although it may sometimes be necessary to repeat attacks against targets at later times. The details of the mathematical formulation and decomposition of this highly complex and interrelated planning, scheduling, and routing problem can be found in Section 5.1.

The desired operational performance improvements that will be realized through the employment of our proposed distributed planning and execution system relate to improvements in accomplishing damage to specified target systems in a manner that:

- Optimally allocates capacitated resources among targets in a manner that reflects Commander's Intent.
- Satisfies operational requirements, such as timeliness of action.
- Reduces overall time to achieve campaign objectives as specified in the Commander's Intent.
- Minimizes operational costs, including flight hours, weapons use, and importantly, attrition.

Our system should enable commanders at the wing operations level and at the strike package level to plan and execute integrated air operations in shorter time frames and with greater effectiveness than is now possible. The system architecture uses information as it becomes available and ensures that local command decisions contribute toward optimization of enterprise-wide strike objectives. The system produces an executable set of integrated tasks for strike aircraft assets that automates and optimizes the use of resources, responds quickly to new information, supports distributed decision-making, and enables Commanders to generate, execute, and adapt strike operations well within the decision cycle of potential adversaries.

Our approach to optimizing distributed air operations planning and execution is to apply control and optimization theory, including hierarchical decomposition and large-scale optimization. Our intent is to provider faster and more effective responses than are now possible to events, including time-critical targets, weather changes, changes in opposing air defense laydown or strategy, aircraft attrition, and changes in force apportionment and campaign strategy. The operational concept for our objective system is to provide decision support for distributed C^2 nodes for strike planning and execution that provide rapid, automated generation of coordinated tasking that is responsive to:

- Commander’s Intent as expressed by geographic and functional target category priorities.
- Commander’s risk management constraints.
- Target characteristics including:

- Weaponeering.
- Reachability with respect to refueling and threat avoidance routing.
- Target dynamics--motion, time criticality, and repair or reconstitution.

- Pilot and unmanned combat air vehicles (UCAV)-unique characteristics, such as risk aversion and in-flight retasking aversion.

Repetitive and tedious, low-level interactions will be eliminated by automation, although operators, through an appropriate interface (not developed under the current effort), will be able to modify plan outputs or initialize plan inputs with low-level specifications as desired. Our system will provide allocation, assignment, and scheduling for formulating executable air asset tasking, including routing for threat avoidance, strike package selection, and weaponeering selection.

Air operations plans are driven by Commander's Intent, where Commander's Intent reflects:

- ***Time***: importance of campaign phase (time phase importance or target time criticality).
- ***Geography***: importance of region.
- ***Target Class***: importance of target class or grouping of targets.
- ***Risk***: the importance of achieving objectives vs loss of resources (including human).

As described in detail in Section 4.1, Commander's Intent is captured in the "Commander's Priority Input Matrix." The information contained in the matrix is mapped into target values that vary with region, by time, by class--along with a trade-off of the benefit of successfully killing the target vs the cost of success. Although it is likely to evolve in future work, this approach is intended to illustrate that Commander's Intent can be captured and translated in order to define objectives for the planning problem.

As reported in later sections of this report, we have performed extensive experimentation in simulated campaigns in order to determine and evaluate the performance and robustness of our closed-loop planning and execution system. The objective of those experiments is to show: (1) that the use of automation allows for faster decision cycles, which results in significant improvement in the effectiveness and timeliness of the overall air campaign; and (2) that closing the loop at high rates to monitor the progress of plan execution and the state of the air operations environment results in robustness to errors in models employed by the algorithms.

To summarize, the scope of the air operations planning problem that we address includes:

- Deciding which targets to hit and when.
- Deciding which assets to use to deliver weapons.
- Deciding routes to follow, refueling, and assembly points.
- Assigning wild weasel and jammer escorts.
- Respecting the laws of physics, logistics, and human performance.
- Accounting for risk in decisions.

The objective is to accomplish damage to specified target systems in a manner that:

- Minimizes operational costs including attrition.
- Satisfies operational requirements such as timeliness of action.
- Allocates capacitated resources between targets in a manner reflecting Commander's Intent.

Major assumptions include:

- Campaign objectives express Commander's Intent.
- Estimates of the state of the air strike resources and their operational environment are provided to the controller at regular intervals.

4 Technical Approach

We employ an approach to decomposing and executing large-scale decision-making problems for dynamic environments that combines the theories of decomposition of large-scale optimization problems and distributed control. Unlike ad hoc approaches to decomposing large-scale operational problems, our approach results in a distributed system for which the problem-solving and decision-making within each distributed C^2 node addresses enterprise-wide objectives. This approach to decomposition both provides significant insight into the nature of the feedback required to close the loop around each of the C^2 nodes within the decomposed problem, and defines the dynamics of the interactions among the control nodes in solving the enterprise-wide problem, including the objectives passed from superior nodes to subordinate nodes and the feedback/status passed from subordinates to superiors.

Real-time, closed-loop optimal control of large-scale dynamic systems (enterprises) remains a challenging problem.[1] We have developed an approach to problems of this class that employs a distributed, multilevel control architecture wherein planning and execution are decomposed to accommodate the near- and far-term impacts of plant disturbances and modeling uncertainties. The decomposition is based on the theory of multi-level optimization for large-scale systems.[2] The structure of the decomposed solution to the optimization problem obtained from this theory forms a basis for our controller architecture as well. In addition to planning, the controller architecture includes execution management, monitoring, and diagnosis at each level. We employ a decomposed formulation for the large-scale military air operations optimization problem described in Section 3.[3]

Our approach to closed-loop, hierarchical control of military air operations employs a distributed control architecture that addresses disturbances at multiple levels of a hierarchically-decomposed planning and execution system to accommodate the near- and far-term impacts of those disturbances. The method we use to decompose the enterprise-wide optimization problem employs the theory of large-scale decomposition that addresses shared objectives and highly constrained resources.

This approach provides a mechanism to assign local autonomy for distributed command and control elements that is consistent with the objectives and constraints established by superior elements, thereby allowing the preservation of traditional command and control organizational structures where desired.

The rest of this chapter is organized as follows. Section 4.1 describes the approach that we have developed for capturing commander's intent and translating it into objectives and constraints for the air operations strike problem. An overview of the technical challenges related to decomposing large-scale problems is presented in Section 4.2. Given a decomposition, we are faced with the problem of defining the closed-loop system architecture that allows the command and control system to both execute the strike plan and receive and act on feedback to replan all or parts of the strike plan. Section 4.3 introduces the system architecture that we employ for that purpose. In Section 4.4, we provide an overview of the theory of multilevel decomposition of large-scale optimization problems, and in Section 4.5, we describe the elements of the simulation that we have developed for evaluating our air strike operations command and control system and provide the rationale for the levels of fidelity we have employed in that simulation. Section 4.6 contains an overview of the interfaces that have employed between the simulation and the closed-loop command and control system, and Section 4.7 outlines the operational scenario that we have simulated and employed in the evaluation of our closed-loop command and control system.

4.1 Modeling Commander's Intent

4.1.1 Introduction

The JFACC process is intended to produce and execute Air Operations plans that accomplish specified damage objectives against specified target systems in the most *effective* manner possible given constraints on knowledge, resources, and policy. The *effectiveness* of Air Operations may be described in terms of:

- Timeliness of action to achieve a desired target-damage end state.

- Minimizing operational costs, including attrition, ordnance cost, and other logistical costs to achieve the desired end state.
- Minimizing uncertainty to end-state and time objectives.
- Maximizing secondary objectives, such as the ability to respond to contingencies.

The "state of the world" during the planning and execution of the air campaign is dynamic in several contexts. With respect to adversary forces:

- New targets and new target locations are discovered and identified during operations.
- Time-sensitive targets may enter a hide state or cessation of critical activities.
- Targets previously attacked may be repaired or reconstituted or battle damage assessment (BDA) may indicate incomplete effects.
- Air defense threats may attrit, relocate or change tactics.

With respect to own forces:

- Aircraft and air crews may be staged into or out of theater.
- Aircraft may suffer damage, attrition, or incur maintenance downtime.
- Bases and supporting infrastructure may suffer temporary or permanent outages.

Ordnance stocks may exhibit scarcity in numbers or distribution of certain items. Finally, there are additional drivers of system dynamics including:

- Physical environment affecting air operations, such as time evolution of weather.
- External time requirements for campaign phasing relating to other warfighting components.
- High-level redirection in response to collateral effects, losses, or execution timeline discrepancies from plans.
- Schedules for supporting processes such as intelligence processing and dissemination.

At a high level of description, capacitated resources, including aircraft, air crews, bases, fuel, and munitions must be allocated, assigned, tasked, and controlled to respond to and to seize opportunities in the dynamic environment. At a more detailed level, the Air Operations plan tasks individual aircraft to carry particular weapon types to particular target release points; to form into packages with strikers and escorts; and depending on the threat environment, to fly particular threat-evading routes and rendezvous for aerial refueling or at assembly points for dynamic package formation. The entire set of taskings is scheduled over a time horizon that may include several successive missions for each aircraft, and the scope may include a number of bases and air warfare functions. Tasking of missions in ground preparation as well as those already dispatched may be altered in response to new information.

As can be anticipated, the Air Operations problem is complex in terms of combinatorial possibilities with decisions of *which* targets to hit and *when* and *how,* allowing many degrees of freedom. In general, there are multiple subobjectives competing for resource tasking, and not everything can be accomplished at once or perhaps at all. The Draper technical approach attempts to construct an Air Operations plan that optimizes, within constraints, an objective function that reflects quantified valuations of target-damage objectives. Enterprise objectives for the entire air campaign are expressed in a compact set of data tables that we have invented and labeled "JFACC *Commander's Intent"* tables. This construct "drives" both resource allocation, detailed planning and plan execution decisions to maximize the accomplishment of the Commander's end-state objective for the enterprise mission while honoring constraints and guidelines.

The minimum elements that are necessary to express Air Operations enterprise objectives include the following:

- ***Time***: importance of campaign phase (time phase importance or target time criticality).
- ***Geography***: importance of region.

- ***Target Class***: importance of target class or grouping of targets.
- ***Risk***: importance of achieving objectives vs loss of resources (including human).

The numerical values from JFACC Commander’s Intent tables are applied to a functional transformation that yields a relative valuation for each target at each time as a function of the overall target damage state. This is central to planning algorithms that task and schedule resources to maximize the aggregate objective function as well as to execution-monitoring algorithms that control dynamic replanning. It is anticipated that the Air Operations staff would customize the data in these tables based on the JFACC Commander's guidance and stated numerical objectives. The Commander’s Intent tables include the following tables, where all entries are indexed by geographic region and target functional category:

- A table that expresses relative target valuation.
- A table that expresses probability of damage objectives.
- A table that expresses target time-perishability or criticality time constants.
- A table that expresses target time-reconstitution time constants.
- A table that indicates the degree of target coupling.

Enterprise objectives are expressed as relative weightings of different target categories and regions. Changes in objectives with campaign phase are expressed by defining a set of times at which the different phases are to commence and replicating the set of tables with values customized to each campaign phase. Overall, the set of tables is a relatively compact representation of enterprise objectives with many values remaining set to default values. The tables and the underlying functional model will be discussed in detail in the following section along with the motivation for formulating the model in that manner. The target damage model also provides an aggregate performance metric that may be quite useful for higher-level decision-making. For example, transitioning from Phase 1 to Phase 2 of a campaign may require the JFACC to produce a 75% reduction in armor and field artillery units. This objective is desired at a specified time as coordinated in the larger plans of the theater or task force commander. Given a set of resources (deployed aircraft) to accomplish this objective, the JFACC desires plans that achieve the objective target damage in the required time and with acceptable costs. Ordinarily, a set of alternative options or plans are desired, and the JFACC will choose from among the options and or request adjustments or additional options from the Air Operations planning staff. The predicted outcome for different plans might be of the form shown in Figure 4.1-1:

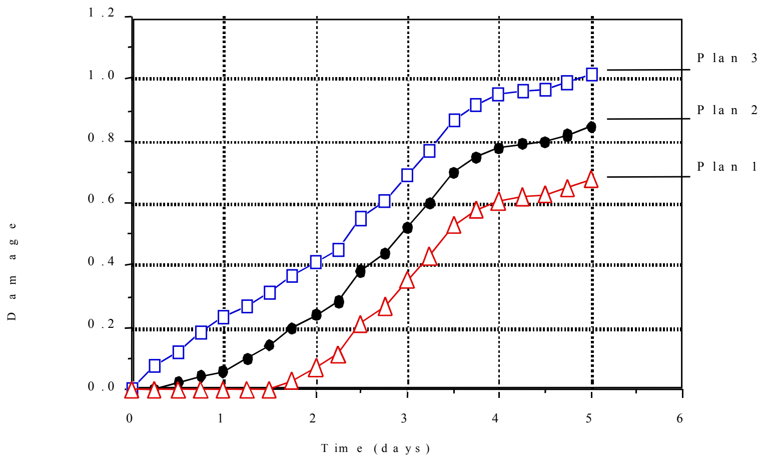


Figure 4.1-1. Damage Metric vs Time for Several Plan Options.

In this example, the predicted performance of the nominal or recommended plan is represented by the middle curve, with the surrounding curves representing alternative plans with different costs and risks. The cost versus time decision space relating to these plans is illustrated in Figure 4.1-2. The cost includes the cost of attrition, the cost of operations, and the cost of munitions, using data from Air Force cost-estimating documents (e.g., AFI 65-503). Although the JFACC is primarily concerned with awarding the loss of aircraft and air crews, the formulation in terms of total dollar cost reflects secondary but very important logistical considerations, including the calculus that losing a $1 billion B-2A is in a different category from losing a $31 million F-15E. It also reflects the constraints that very effective weapon systems that happen to have very expensive unit acquisition costs are typically procured in very limited numbers.

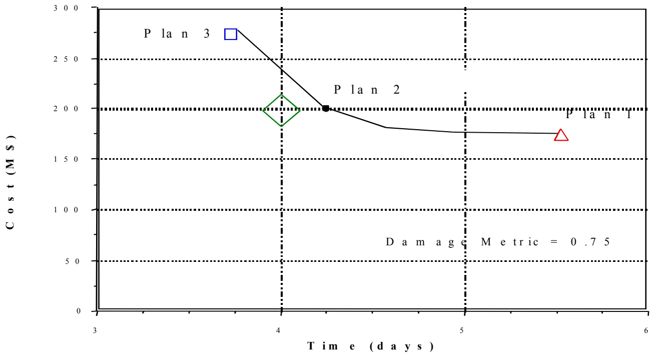


Figure 4.1- 2. Cost vs Time of Plan Options that Achieve Specified Objectives.

The nominal end state desired by the Commander, the Commander's Intent, is indicated by the diamond corresponding to 4 days campaign time and $200 M in cost. The nominal plan misses the time of accomplishing the damage objectives by 0.25 days. To address this planning shortfall, the JFACC might do any of the following:

- Try to negotiate a 4.25-day phase transition time with the theater commander.
- Relax the cost constraint and assume more risk in order to meet the 4-day timeline.
- Allocate more resources to meet both desired time and cost constraints.

Ordinarily, the objective damage metric remains a constant, while different plan options trade cost and risk versus time to achieve that objective. If none of the options are acceptable, it is conceivable that the damage objectives could be renegotiated with the theater commander, a process that is outside the scope of JFACC Air Operations Planning.

The target value model based on Commander's Intent is at the core of the optimization of Air Operations plan generation and execution. The details of that model will be described in the next section.

4.1.2 Commander's Intent-Based Target Value Model

The Air Operations planning problem involves many targets, each of which has a set of attributes that include:

- Functional category (e.g., integrated air defense, command, control, communications (C^3); lines of communication; petroleum, oil, and lubricants (POL); maneuver units).
- Geographic location.
- Time perishability.
- Current damage state and planned damage state.

At each stage in the air campaign, capacitated resources will limit the number of targets that can be attacked in a given time frame. The essence of optimization as applied to Air Operations planning requires a method for identifying which targets are more "valuable" than other targets, including the notion that target value may have dynamics, valuation may depend on the current damage state, and targets may have valuation as part of a "target system." This notion of relative valuation pertains only to the enterprise objective and never assumes the dimensions of costs that we associate with our own operations.

In practice, there are many analytical sources of target nomination, including different ISR systems, different organizational constituencies, and even different service branches. Different categories of targets have different importance to different constituencies. In addition, the target list will contain targets that were submitted at different times with priorities shifting over time. As a result of these factors, the goal of establishing normative target values at the point of target nomination has proved to be intractable.

To compare targets on the diverse list at any time, an analytical function is defined that captures the essence of target valuation for Air Operations planning. The JFACC Commander's Intent tables provide numerical parameters for this function that are used along with attributes of the set of target objects to produce an evaluated value for each target.

The first issue to be addressed is the relative weighting of different target functional categories and geographic regions. The commanders guidance is concretely expressed as a table of weightings with rows varying over target functional categories and columns expressing geographic regions or groupings. Table 4.1-1 shows an example.

Table 4.1-1. Example Commander's Intent Input Matrix: Campaign Phase 1.

Functional Category	Western AD District	Northern AD District	Southern AD District	WC Invasion Salient	Commanders Special Grouping
Integrated Air Defense C^3	10	30	20	10	
Integrated Air Defense non-C^3	5	10	5	10	
Other C^3	10	20	5	10	
Lines of Communication	1	20	1	10	
Petroleum, Oil & Lubrication	1	5	1	5	
Maneuver Units	1	5	1	10	10
Long Range Artillery	10	10	5	20	10

The number of functional categories and the number of geographic regions is scenario-dependent and the numbers in individual table cells, the concatenation of the category and region specification, is interpreted as a dimensionless, relative value. For example, targets in the Integrated Air Defense C^3 category of the Northern Air Defense District in Table 4.1-1 are considered to be three times as valuable as targets in the Lines of Communication category of West Cyberland Invasion Salient region, all other factors being the same. The last column, Commanders Special Grouping, provides an additional global weighting of targets that is independent of region. Such a construct may be useful for implementing the Commander's guidance indicating a priority to destroy targets in a particular category regardless of where they are located in the theater of operations.

For a campaign phase that is primarily concerned with establishing air superiority, for example, the categories of airfields and any category relating to air defenses might be accorded especially high values. The variation between geographic regions can be used to apply economy of force when resources are inadequate to globally accomplish particular phase objectives. It should be noted that the weighting of any individual "cell" at a particular time (campaign phase) is determined by normalizing with respect to all the numbers in the table. In Table 4.1-1, Northern AD District Integrated Air Defense C^3 weighting is 30/271, or about 11% of the overall effort. In general, there is a distribution of effort between competing objectives that emphasize different cells. If the desire is to focus 50% of the effort on the Northern AD District Integrated Air Defense C^3 targets, then the value in that cell must be increased to 241, or the values in the other cells must be reduced accordingly.

One method for the Air Operations staff to choose values that reflect Commander's Intent is to set up a system that maps a few discrete values, say 0, 1, 5, 10, 20, and 30, to a few qualitative descriptions of overall importance to the enterprise mission for this campaign phase such as "not at all," "nominal," "somewhat," "important," "very important," and "of the highest importance." The important point is that continuous values cannot be established readily by human qualitative judgement. Human operators can be trained to use a discrete mapping, and the system can be made relatively insensitive to the expected dispersion in qualitative value assessment between different operators in different scenarios and at different times.

The next set of issues to be addressed in the target valuation function is the notion of target hardness or resiliency, the interdependence of individual targets in target "systems," and the definition of threshold levels for target damage effects that relate to weaponeering decisions. These issues are all interrelated. An illustrative example is a scenario with 10 air defense targets. The imposition of 50% damage may have a different valuation depending on whether all 10 sites experienced a 50% damage level or whether 5 sites were completely destroyed leaving 5 sites completely intact. Since there is likely to be excess capacity or redundancy in the air defense system as a whole, the situation with 5 sites remaining intact may be more threatening to air operations than the situation where all 10 sites have experienced 50% damage. Of course, if each site has a high level of redundancy, then it may be necessary to specify even higher damage levels for each site to significantly degrade the threat to air operations.

Another example of target interdependency or target "systems effects" is the scenario with multiple bridges across an otherwise inaccessible terrain. To interdict maneuver forces or resupply across that terrain, it is insufficient to destroy one or two bridges if there are alternative lines of communication. If there is excess capacity in the lines of communication, there may be very little benefit to attacking any individual bridge. Significant target damage value is not obtained until a substantial fraction of transport capability is interdicted, implying that it may be necessary to drop several bridges in the immediate vicinity.

Models for the effects of target hardness and target interdependence are implemented by the definition of tables for specifying threshold damage levels to individual targets and the specification of a target system payoff function. These tables, as before, are indexed by target functional category and by geographic region. Targets do not contribute any value to the target damage function until the fractional damage level exceeds the threshold damage level, Pd, specified for the table cell containing that target. Weaponeering data for each target are used to select weapons, aircraft, and packages to achieve the specified threshold damage levels. The value of Pd is defined as the probability of achieving a specified damage effect that ranges from temporary incapacitation, through partial damage, "mobility kill," to total destruction. For an area target, Pd can be interpreted as either the probability of damaging (to a specified level) a single target randomly located on the target area or as fractional damage, the expected fraction of target elements in the target area that are damaged to the desired level. The particular effect and level desired depends on the functional category and is specified by the Commander's guidance.

The payoff function is evaluated for each category/region cell of the table and may be linear for targets with independent valuation or nonlinear for targets with collective valuation. For target systems that are operating near capacity limits, small increments of damage can have superlinear effects on reduction of capacity. Examples are nonlinear effects on communication, transport, or any network system that is operating near its capacity limit. On the other hand, target systems that contain substantial excess capacity or redundancy may exhibit threshold type effects where functionality is maintained for considerable fractional damage until the damage level exceeds a specified threshold. A functional form that captures the nonlinear behavior in parametric form is:

$$f(\beta;\alpha) = 1 - \alpha^{\left[\frac{(1-\alpha)\,\beta}{(1-\beta)}\right]}$$

where

β = fraction of targets in cell with $P_{D|AS}$ exceeding specified threshold

α = 0.10 for superlinear collective behavior

α = 0.75 for threshold collective behavior

Figure 4.1-3 depicts the variation of f(β;α).

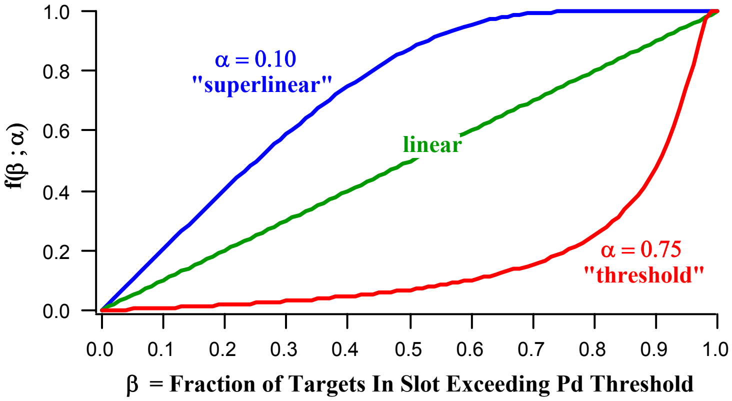


Figure 4.1-3. Category/Region Cell (Slot) Payoff Functions.

Since the variable β is defined as the fraction of targets in a category/region cell with Pd exceeding the specified threshold, the addition of targets to a cell that are not yet in a plan will reduce the collective payoff function for that cell. For a linear payoff function, targets addressed with Pd exceeding the threshold will add independently of each other. For cells specified with a superlinear payoff function, the payoff slope is steepest for the initial targets that are addressed with Pd exceeding the threshold. Cells specified with the "threshold" payoff function require that a large fraction of targets in the cell be addressed with adequate Pd before substantial payoff is achieved. It should be noted that:

- There is no attempt or requirement to specify relative target values within a cell (i.e., all targets successfully addressed have the same weighting).
- Target value is accrued only for estimated damage exceeding a specified threshold. In other words, targets only have value on specific weaponeering choices.

The specification of three models for the payoff function captures the qualitative features of the problem domain without imposing the analytical requirement on the Air Operations planning staff to specify a continuous value for the α parameter.

The target dynamics models and the JFACC Commander's Intent tables pertaining to target time criticality and reconstitution are described in the next section.

4.1.3 Modeling the Achievement of Commander's Intent

The formula for applying weaponeering estimates of probability of damage to the current target damage state to obtain the post-strike estimate of probability of damage is

$$P_{D|AS} = 1 - [1 - P_{D|W}][1 - P_{D|CS}] \qquad (1)$$

where

$P_{D|AS}$ = estimate of the probability of damage to the target after (as a result of) the strike

$P_{D|W}$ = estimate of the probability of damage to an undamaged target as a result of the use of N_w weapons

$P_{D|CS}$ = estimate of the fractional damage of the target

The number of weapons that are required to achieve a specified damage level is independently specified for each target from weaponeering analysis that is presumed to occur as part of the target development and nomination process. The weaponeering specification indicates how many weapons of a particular type are required to be delivered on "virgin" targets to achieve the Pd threshold. If a different number of weapons, N, are released on a target, the $P_{D|W}$ factor is replaced by the factor P_{DN} obtained by logarithmic interpolation:

$$P_{D1} = 1 - [1 - P_{D|W}]^{1/N_w}$$

$$P_{DN} = 1 - [1 - P_{D1}]^N$$

The models for target dynamics include the effects of time perishability or criticality and the effects of target repair or reconstitution. Time perishability of target value reflects the fleeting intrinsic value of striking a target after a high value activity has transpired, as well as the diminished ability to acquire and damage a fleeting target successfully. A simple functional manner to incorporate this effect is to append a first-order, exponential time-decaying factor to Eq. (1) of the form

$$exp[- (t_s - t_0)/\tau]$$

where

t_0 = reference time (often the time of entry onto the target list)

t_s = scheduled strike time

τ = characteristic time

The time constant τ is specified by target functional category and geographic region in one of the JFACC Commander's Intent tables. The specification by the standard tabular structure possibly replicated for different campaign phases, asserts that all targets in the target functional category and geographic region cell can be described by the same parameters for time criticality, and removes the burden of specifying these data for each target individually. The exponential decay factor is applied to the valuation that would otherwise be obtained for that target.

The effects of reconstitution of a target can be expressed in precisely the same form, as

$$exp[- (t - t_s)/\tau_r]$$

where

t = current time

t_s = strike time

τ_r = characteristic reconstitution time

Targets with $P_{D|CS}$ modified by these factors will tend to fall below the required Pd thresholds (if not addressed in a timely manner by the controller), and will in turn reduce the respective cell payoff functions, and hence, the weighted aggregate target damage metric. An example of target reconstitution is the repair of damaged lines of communication, filling in holes in runways, etc. The time constants may range from hours to days leading to significance of this effect over multiday campaigns.

An aggregate target damage metric is generated by:

- Evaluating the post-strike $P_{D|CS}$ for each target.
- Evaluating the fraction β of targets that exceed the specified Pd threshold in each cell.

- Evaluating the payoff function using the payoff model specified for each cell.
- Multiplying the payoff value by the Commander's Intent input matrix weighting for each cell.
- Summing over all cells.
- Normalizing by the sum of all weighting values.

The aggregate damage metric is constrained to lie within the range (0,1), a convenient property for comparison and ready interpretation of results across scenarios, scenario perturbations, and experiments with different planning methods. It represents the aggregate *fractional* damage of all *known* targets with weighting according to Commander's Intent input. As new targets become known, the *fractional* damage decreases, of necessity, until those targets are addressed.

For those category-region cells that are devoid of targets, the logical possibilities for the assignment of the fraction β exceeding the Pd threshold are 0 or 1. The former case penalizes the aggregate damage score simply because there are no targets in particular cells whereas the latter gives full credit for those cells that are devoid of targets. Both of these effects result in a compressed dynamic range for the damage metric, the former with a reduced upper limit and the latter with a nonzero lower or initial limit. Alternatively, a more aesthetic compensation that preserves the full (0,1) range for the damage metric is to adjust the normalization of the aggregate target damage to remove the weighting of those cells without targets.

As targets are discovered during operations and are added to the set of targets, the population of individual category-region cells can increase over time. As a result, the fraction β of targets damaged beyond the Pd threshold for a cell will instantaneously decrease on the addition of new targets to that cell. This nonmonotonic property of the aggregate damage metric is a consequence of the physical model of cells as target systems with payoff functions that key on the fraction of targets in each cell that are damaged beyond a specified threshold. Aggregate damage can also decrease as a result of target reconstitution. Finally, the aggregate metric can exhibit discontinuities at campaign phase boundaries when Commander's Intent input weightings change between phases. It does not appear possible to preserve the essential normalization property if changes are made to enforce monotonicity. The nonmonotonic behavior and discontinuities at phase boundaries are illustrated in Figure 4.1-4. It should be noted that alternative metric formulations that enforce monotonicity and continuity would alter the relative normalization and destroy the ability to compare the different cases on a common scale.

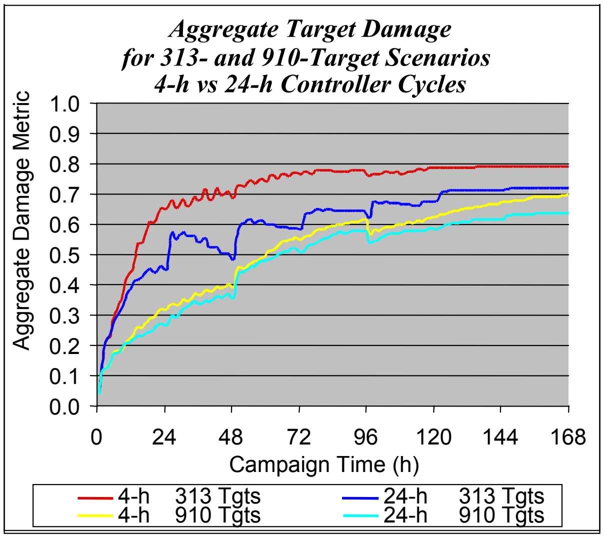


Figure 4.1-4. Aggregate Damage Rate as Function of Time.

The salient points for analysis are the time to achieve specified values of the aggregate target damage metric, the asymptotic value achieved, and the variation of these features with planning parameters, such as the time interval between major replans, the risk aversion parameters, and the cost to replan. Variations of these features with physical problem perturbations are also interesting, such as the number of targets relative to the number of aircraft resources, and hence, system capacity, the variation with the fraction and time constants of time critical targets, and the variation with the use of UCAVs with lower costs, higher risk parameters, and lower retasking costs.

The jitter that appears in the aggregate damage metric arises from the wavelike structure of mission packages that are dispatched with a time that reflects mission flight time and ground turnaround time, as well as new target discovery at arbitrary times. The former gives rise to crests in the waves and the latter to depressions in the waves. If we perform a least squares curve fit across the jitter and the phase discontinuities, it is then possible to invert the fitted curve to systematically determine the time to achieve various levels of aggregate damage. The result is illustrated in Figure 4.1-5, which shows the time saved by reducing the loop closure interval from 24 to 4 hours as a function of the objective damage level and parametrically with different physical perturbations.

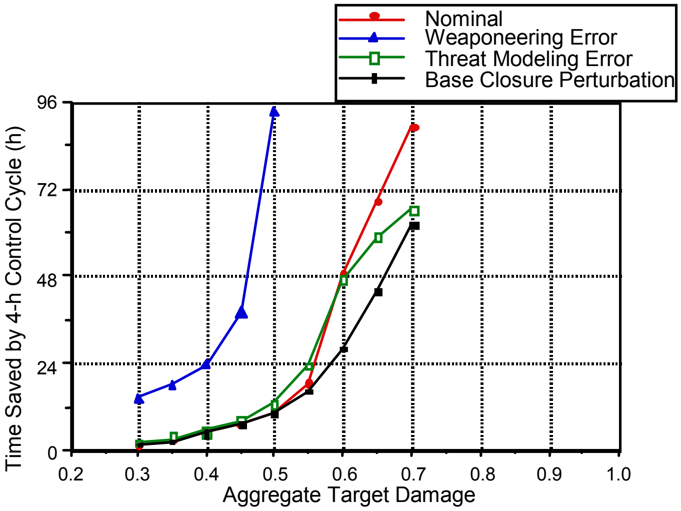


Figure 4.1-5. Campaign Time Saved by Reducing Loop Closure Interval from 24 to 4 h.

4.1.4 Target Value Summary

The parameters that determine target value include the following set of parameters defined for *each* functional category/region cell:

- Commander's relative weighting.
- Number of targets.
- Pd threshold.
- τ and τ_r characteristic times for perishability and reconstitution.
- Specification of superlinear, linear or threshold payoff function.

The number of targets in each cell is evaluated by incremental counting of targets as they are added to the target queue. Assuming that each of these parameter items is represented by a two-dimensional table (rows are categories and columns are regions), there are four tables to be entered for each campaign phase, although as a practical matter, only the Commander's relative weighting may require separate entries for each phase. The set of information that needs to be defined for each target includes the following with only the **bold-faced** items required for the target valuation model:

- BE (basic encyclopedia) number.
- Name.

- **Region.**
- Country.
- Latitude, longitude, altitude.
- **Functional category.**
- Graphic reference.
- **Current damage level or Pd.**
- **Begin availability.**
- Vector of munition type**,** number of weapons**, $P_{D|W}$ for each weaponeering solution.**

The accrued "damage value" contribution of a target depends on the post-strike Pd exceeding the specified Pd threshold for that cell, the same test for all other targets in the cell, on the specified type of payoff function for that cell, and finally on the weighting factor for that cell from the Commander's Intent input matrix. The weightings are normalized by the sum over all cells, resulting in an aggregate damage metric that is always normalized in the range (0,1), but that exhibits nonmonotonicity and discontinuities across campaign phase boundaries.

4.1.5 Air Operations Plan Review

Table 4.1-2 contains a summary output of an example air operations plan that is to be reviewed for acceptance by the Commander either at the theater level (all packages) or at the subproblem level (only those packages relevant to the subproblem. The table lists each of the assigned aircraft missions, with grouping by target and with ordering by time on target. The information includes:

- Target ID.
- The target functional category (LOC = Lines of Communication, IAD_NONC3 = Integrated Air Defense, Non-Command Control and Communications, POL = Petroleum, Oil and Lubricants) and the specified Commanders Intent input weighting value.
- The target geographic region (NORTH_ADD = Northern Air Defense District, WC_SAL = West Cyberland invasion Salient).
- The aircraft ID with base designations EAGLE = Gaan, HAWK = Tuljeti, TIGER = Mt Hegan, BOOMER = Guam, BANGER = Darwin.
- Standard conventional load (i.e., mission configuration).
- Package role of that aircraft.
- Time on target.
- Mission flight distance in kilometers.
- Mission endurance in seconds.
- Estimated mission attrition risk.

The table allows the Commander to review the air operations plan to determine whether intent is indeed being met, and possibly to make suggestions to, for instance, the controller to include or exclude specific targets from the plan. It can also be seen that a variety of packages are utilized, with weasel and jammer resources employed to reduce mission risk for those targets requiring passage through heavily defended areas that can not be totally avoided by routing. The number of strikers in a package and the standard conventional load will depend on the input target weaponeering specification. Not explicitly shown are the aerial refueling activities. Since the maximum endurance of the tactical strike aircraft is around 9000 s, it can be inferred that there are several refueling hops for those aircraft on long endurance missions to remote targets in the Western and Southern Air Defense Districts.

It should also be noted that most of the packages travel in formation, but that for a number of targets in the middle of the table, e.g., around Target 1183, there is evidence of retasking and dynamic package formation.

Table 4.1-2. Summary Air Operations Plan Report for Commander Review and Feedback.

Target ID	Target Functional Category/ Weighting Value	Target Geographic Region	Aircraft ID	Standard Conventional Load	Package Role	Time On Target (s)	Mission Distance (km)	Mission Endurance (s)	Estimated Mission Risk
1981	LOC	WC_SAL	EAGLE-1-5	F-4-031	weasel	2894	90	569	0.0000000
	10		EAGLE-3-7	F-5-010	striker				
1963	LOC	WC_SAL	EAGLE-3-8	F-5-010	striker	2988	133	752	0.0000000
	10		EAGLE-1-6	F-4-031	weasel				
1949	LOC	WC_SAL	EAGLE-3-1	F-5-005	striker	3157	211	1093	0.0000000
	10		EAGLE-3-0	F-5-005	striker				
1901	LOC	WC_SAL	EAGLE-3-11	F-5-010	striker	3593	414	1967	0.0000000
	10		EAGLE-1-9	F-4-031	weasel				
3229	AIRFIELDS	SOUTH_ADD	TIGER-1-4	F-4-031	weasel	4000	602	2781	0.0000000
	5		TIGER-3-6	F-5-010	striker				
2943	LOC	WC_SAL	EAGLE-1-1	F-4-031	weasel	4144	669	3068	0.0000000
	10		EAGLE-3-3	F-5-010	striker				
2005	LOC	WC_SAL	EAGLE-1-3	F-4-031	weasel	4403	789	3587	0.0000000
	10		EAGLE-3-5	F-5-010	striker				
2645	IAD_NONC3	NORTH_ADD	TIGER-1-2	F-4-031	weasel	4514	840	3808	0.0000000
	10		TIGER-3-4	F-5-010	striker				
3049	IAD_C3	SOUTH_ADD	TIGER-3-1	F-5-005	striker	4523	844	3827	0.0000000
	20		TIGER-3-0	F-5-005	striker				
171	IAD_C3	SOUTH_ADD	TIGER-3-3	F-5-005	striker	4526	846	3833	0.0000150
	20		TIGER-3-2	F-5-005	striker				
			TIGER-1-0	F-4-031	weasel				
			TIGER-2-0	F-2E-033	jammer				
			TIGER-1-1	F-4-031	weasel				
2321	LOC	NORTH_ADD	HAWK-2-0	F-2E-033	jammer	4581	871	3941	0.0004465
	20		HAWK-1-0	F-4-031	weasel				
			HAWK-3-0	F-5-010	striker				
2333	LOC	WC_SAL	EAGLE-3-14	F-5-010	striker	4750	949	4280	0.0000000
	10		EAGLE-1-12	F-4-031	weasel				
2043	LOC	WC_SAL	EAGLE-3-15	F-5-010	striker	4784	965	4347	0.0000000
	10		EAGLE-1-13	F-4-031	weasel				
1995	LOC	WC_SAL	EAGLE-1-4	F-4-031	weasel	4788	967	4357	0.0000000
	10		EAGLE-3-6	F-5-010	striker				
2929	OTHER_C3	SOUTH_ADD	TIGER-1-3	F-4-031	weasel	4918	1027	4615	0.0005112
	5		TIGER-3-5	F-5-010	striker				
			TIGER-2-1	F-2E-033	jammer				
2523	IAD_C3	SOUTH_ADD	TIGER-1-5	F-4-031	weasel	5042	1084	4864	0.0008851
	20		TIGER-3-7	F-5-010	striker				
79	IAD_NONC3	SOUTH_ADD	EAGLE-3-17	F-5-005	striker	5122	1121	5024	0.0002998
	5		EAGLE-3-16	F-5-005	striker				

Table 4.1-2. Summary Air Operations Plan Report for Commander Review and Feedback. (Cont.)

Target ID	Target Functional Category/ Weighting Value	Target Geographic Region	Aircraft ID	Standard Conventional Load	Package Role	Time On Target (s)	Mission Distance (km)	Mission Endurance (s)	Estimated Mission Risk
473	LOC	NORTH_ADD	EAGLE-2-0	F-2E-033	jammer	5134	1127	5048	0.0002719
	20		EAGLE-3-2	F-5-010	striker				
			EAGLE-1-0	F-4-031	weasel				
2407	IAD_NONC3	SOUTH_ADD	EAGLE-3-13	F-5-005	striker	5146	1133	5072	0.0000000
	5		EAGLE-1-11	F-4-031	weasel				
1281	AIRFIELDS	NORTH_ADD	EAGLE-3-12	F-5-010	striker	5195	1155	5169	0.0005898
	20		EAGLE-2-4	F-2E-033	jammer				
			EAGLE-1-10	F-4-031	weasel				
3157	AIRFIELDS	NORTH_ADD	EAGLE-3-9	F-5-010	striker	5253	1182	5286	0.0004554
	20		EAGLE-2-2	F-2E-033	jammer				
			EAGLE-1-7	F-4-031	weasel				
103	AIRFIELDS	NORTH_ADD	EAGLE-3-10	F-5-010	striker	5258	1185	5297	0.0004907
	20		EAGLE-2-3	F-2E-033	jammer				
			EAGLE-1-8	F-4-031	weasel				
1459	LOC	NORTH_ADD	EAGLE-1-2	F-4-031	weasel	5348	1226	5475	0.0010924
	20		EAGLE-3-4	F-5-010	striker				
			EAGLE-2-1	F-2E-033	jammer				
325	POL	NORTH_ADD	HAWK-3-1	F-5-005	striker	5984	1520	6747	0.0007773
	5		HAWK-2-1	F-2E-033	jammer				
			HAWK-1-1	F-4-031	weasel				
1183	OTHER_C3	WEST_ADD	HAWK-3-5	F-5-005	striker	8429	2940	13279	0.0019494
	10		HAWK-3-4	F-5-005	striker	8429	2940	13279	0.0019494
			HAWK-3-2	F-5-005	striker	8559	2527	11897	0.0003357
			HAWK-1-2	F-4-031	weasel	8559	2466	11234	0.0001679
2851	OTHER_C3	WEST_ADD	HAWK-3-3	F-5-010	striker	8677	2582	12135	0.0014516
	10		HAWK-1-3	F-4-031	weasel	8677	2521	11471	0.0012837
2463	OTHER_C3	SOUTH_ADD	TIGER-1-7	F-4-031	weasel	9408	2844	12866	0.0011088
	5		TIGER-3-9	F-5-005	striker	9408	2921	13597	0.0012766
2977	OTHER_C3	SOUTH_ADD	TIGER-1-6	F-4-031	weasel	9433	2856	12916	0.0008953
	5		TIGER-3-8	F-5-010	striker	9433	2932	13646	0.0010631
3125	IAD_C3	SOUTH_ADD	TIGER-3-11	F-5-005	striker	9524	2974	13827	0.0015156
	20		TIGER-3-10	F-5-005	striker				
511	AIRFIELDS	WEST_ADD	BANGER-1-1	B-101-007	striker	10130	2190	9640	0.0000000
	10		BANGER-1-0	B-101-007	striker				
1927	LOC	WC_SAL	BOOMER-1-1	B-100-007	striker	14102	4029	17584	0.0000000
	10		BOOMER-1-0	B-100-007	striker				
1839	IAD_C3	WEST_ADD	BOOMER-2-0	B-102-007	striker	15174	4875	21638	0.0000011
	10		BOOMER-2-1	B-102-007	striker				
29	AIRFIELDS	SOUTH_ADD	EAGLE-3-7	F-5-005	striker	15865	1248	5572	0.0006835
	5		EAGLE-1-5	F-4-031	weasel				

Table 4.1-2. Summary Air Operations Plan Report for Commander Review and Feedback. (Cont.)

Target ID	Target Functional Category/ Weighting Value	Target Geographic Region	Aircraft ID	Standard Conventional Load	Package Role	Time On Target (s)	Mission Distance (km)	Mission Endurance (s)	Estimated Mission Risk
1775	IAD_NONC3	SOUTH_ADD	EAGLE-3-8	F-5-005	striker	16248	1341	5972	0.0008842
	5		EAGLE-1-6	F-4-031	weasel				
1623	AIRFIELDS	SOUTH_ADD	EAGLE-3-1	F-5-005	striker	16406	1256	5606	0.0013670
	5		EAGLE-3-0	F-5-005	striker				
2279	AIRFIELDS	SOUTH_ADD	EAGLE-3-11	F-5-005	striker	17177	1209	5400	0.0008258
	5		EAGLE-1-9	F-4-031	weasel				
841	IAD_C3	SOUTH_ADD	TIGER-3-6	F-5-010	striker	17777	1109	4973	0.0016812
	20		TIGER-1-4	F-4-031	weasel				
627	AIRFIELDS	SOUTH_ADD	EAGLE-3-3	F-5-005	striker	18283	1211	5409	0.0008656
	5		EAGLE-1-1	F-4-031	weasel				
731	IAD_C3	SOUTH_ADD	TIGER-2-0	F-2E-033	jammer	18417	919	4149	0.0002235
	20		TIGER-1-0	F-4-031	weasel				
			TIGER-3-0	F-5-010	striker				
3309	AIRFIELDS	SOUTH_ADD	EAGLE-1-3	F-4-031	weasel	18809	1214	5425	0.0009385
	5		EAGLE-3-5	F-5-005	striker				
3191	POL	NORTH_ADD	HAWK-2-0	F-2E-033	jammer	19098	1184	5294	0.0012362
	5		HAWK-1-0	F-4-031	weasel				
			HAWK-3-0	F-5-005	striker				
1249	IAD_C3	SOUTH_ADD	TIGER-1-1	F-4-031	weasel	19223	1292	5762	0.0025017
	20		TIGER-3-1	F-5-005	striker				
1133	AIRFIELDS	SOUTH_ADD	EAGLE-3-14	F-5-010	striker	19510	1218	5439	0.0010055
	5		EAGLE-1-12	F-4-031	weasel				
1689	AIRFIELDS	SOUTH_ADD	EAGLE-1-4	F-4-031	weasel	19587	1218	5441	0.0010160
	5		EAGLE-3-6	F-5-010	striker				
2713	AIRFIELDS	SOUTH_ADD	EAGLE-3-15	F-5-005	striker	19603	1230	5491	0.0012441
	5		EAGLE-1-13	F-4-031	weasel				
255	IAD_C3	SOUTH_ADD	TIGER-3-2	F-5-005	striker	19896	1604	7107	0.0024633
	20		TIGER-1-2	F-4-031	weasel				
685	IAD_C3	SOUTH_ADD	TIGER-3-4	F-5-005	striker	22918	2818	13151	0.0023718
	20		TIGER-3-3	F-5-005	striker				

In addition to reviewing the planned tasking, the Commander can also examine the target damage performance of the executed plans to assess progress and detect problem areas. In this regard, it is probably more useful to examine the individual functional category-geographic region cells rather than the aggregated or "rolled-up" target damage metric. An example of a display that might be useful is shown in Figure 4.1-6, with the target fractional damage in each cell displayed in perspective as a function of campaign time. Deficiencies or anomalies can be identified readily from such a display, and plans adjusted for remediation. Of course, metrics that are aggregated in one dimension, over geographic regions or over functional categories only, can also be generated and displayed for summary or overview displays.

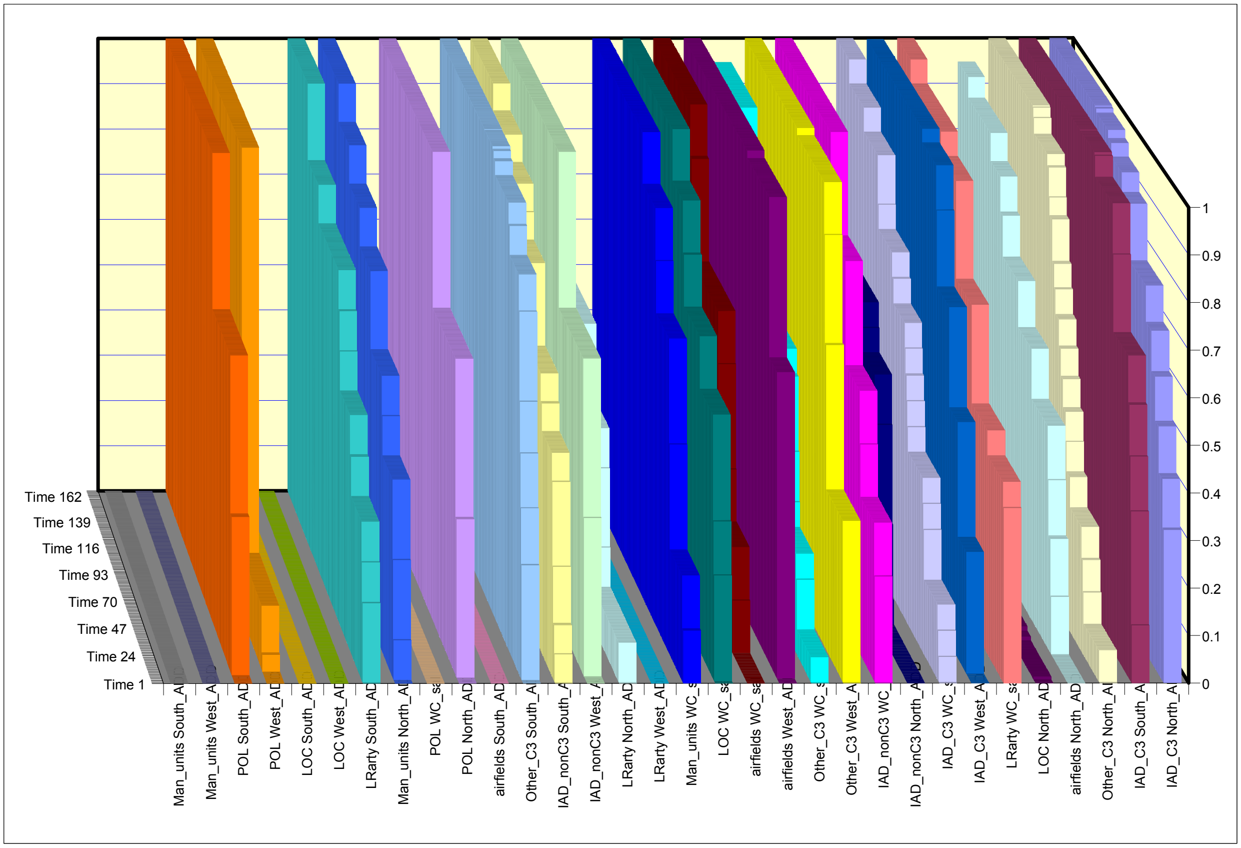


Figure 4.1-6. Damage for 913 Target Case – Satisfaction of Commander's Intent

Figure 4.1-6 can be collapsed to a simple bar chart for examination of the target damage by cell at specified times, such as the currently assessed damage levels or the planned damage levels for operations through D + 2 days. What is most obvious from this figure is the absence of effort for some cells, and secondarily, the different rates at which targets in other cells are addressed. These results are primarily a function of Commanders Intent input weightings for the cells, transit distance from aircraft deployed at different bases, air defense threat laydowns around different targets, and the number of targets and number of weapons required to be delivered on those targets.

Tables and graphics can be generated to display other aspects of the intended Air Operations Plan and execution thereof, including examination of the use of individual aircraft, base, aerial tanker, and munitions expenditures by type. The information contained in these reports can indicate issues that limit sortie generation, such as the distribution of escort aircraft throughout the theater, as well as logistical issues relating to ordnance resupply and availability.

Finally, a graphic that presents useful monitoring information is the plot of sorties versus campaign time shown in Figure 4.1-7. The curve labeled "Sorties in Progress" shows that there are an average of about 20 sorties in progress during the first 72 hours, with fluctuations between about 10 and 40 and a wavelike structure corresponding to planning cycles and typical flight transit time plus turnaround time. Initially, the number of retaskings is relatively large, with most of these changes occurring during ground preparation. The sortie rate decreases as the target queue becomes depleted from air operations and from time-critical targets surviving past their window of vulnerability or criticality.

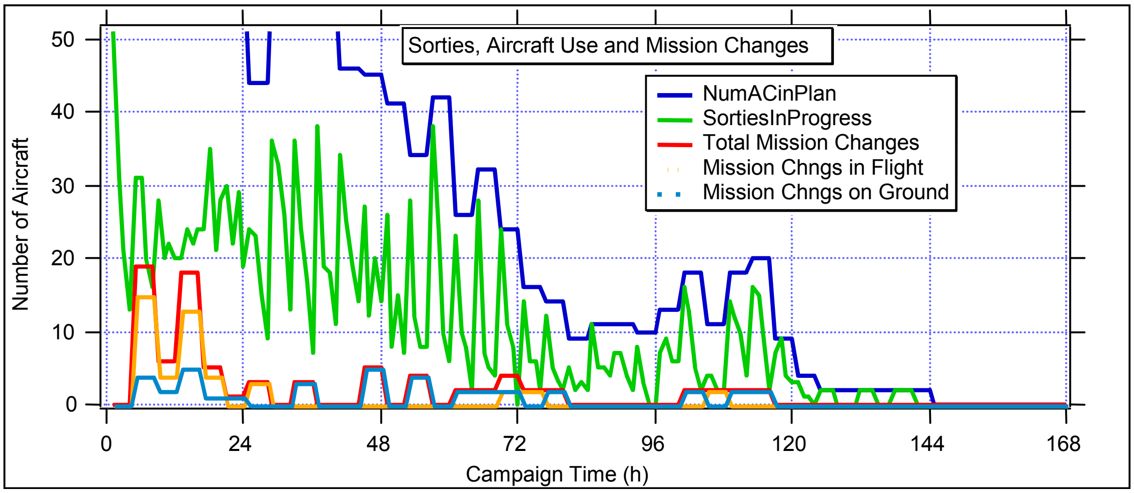


Figure 4.1-7. Sorties In Progress vs Time for 313-Target, 4 -h Case.

4.2 Decomposition

4.2.1 Decomposition and Closed-Loop Control

The objective of multilevel optimization is to decompose a complex optimization problem into a hierarchy of simpler problems while still retaining a global or enterprise-wide view of the problem. Research in the theory of large-scale optimization[1][2] has produced a variety of approaches to decompose very large-scale problems into components or subproblems that are computationally tractable. The simpler optimization problems are solved independently at each level of the hierarchy with the superior or master levels coordinating the solutions of the decoupled subordinate level problems.

The central topic of our effort is the application of these analytical approaches to decomposing large-scale optimization problems to closed-loop control for large-scale enterprise problems with complex objective functions and constraints. The solutions to the subproblems at the lowest levels of the decomposition represent a plan of activities that are to be pursued by the enterprise's physical entities in prosecuting the business of the enterprise, e.g., missions for individual aircraft. At higher levels, the solutions produce objectives and constraints to be employed by successively lower levels, e.g., the allocation of sets of targets to sets of strike packages. The environment (elements that are not controlled, e.g. weather, topography, opposing and neutral forces) and the plant (the system to be controlled) within which those activities are to be pursued are represented (modeled) in the formulation through a variety of constraints. In order to reduce sensitivity to modeling errors and disturbances, we employ feedback providing information about the actual evolution of the state of the environment and the system to be controlled. The following describes the control architecture that we use in closing the loop.

Figure 4.2-1 represents one of the command and control nodes within the controller architecture. Sensing the plant, i.e., the "system to be controlled" and the environment provide feedback. The "system" may be physical entities within the plant that are being controlled, or it may represent an aggregation of lower-level problem solving nodes along with the entities they control. A closed-loop, hierarchical decomposition is a recursive implementation of the node illustrated in Figure 4.2-1, where the "system-to-be-controlled" is one or more subordinate-level processes that are "controlled" or coordinated by an upper Master level, as shown in Figure 4.2-2.

In Figure 4.2-2, the plant for the Master level is an aggregation of the lower-level nodes and their plants. For the results discussed later in this report, the lowest levels control a simulation rather than the actual plant.

The feedback should contain the information required to evaluate progress toward the solution to the subproblem being solved. Since the solutions generated will span a finite time horizon, models will be required to predict future states and status based on the planned course of action and estimates of the current state.

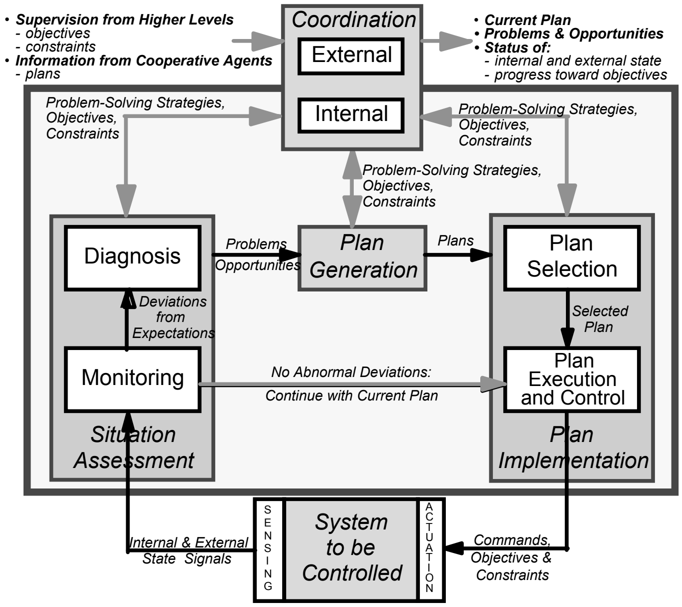

• Supervision from Higher Levels
- objectives
- constraints
• Information from Cooperative Agents
- plans
Coordination
External
Internal
• Current Plan
• Problems & Opportunities
• Status of:
- internal and external state
- progress toward objectives
Problem-Solving Strategies, Objectives, Constraints
Problem-Solving Strategies, Objectives, Constraints
Problem-Solving Strategies, Objectives, Constraints
Diagnosis
Problems
Opportunities
Plan Generation
Plans
Plan Selection
Deviations from Expectations
Selected Plan
Monitoring
No Abnormal Deviations:
Continue with Current Plan
Plan Execution and Control
Situation Assessment
Plan Implementation
Internal & External State Signals
SENSING
System to be Controlled
ACTUATION
Commands, Objectives & Constraints

Figure 4.2-1. Functional Decomposition of a Command and Control Node.

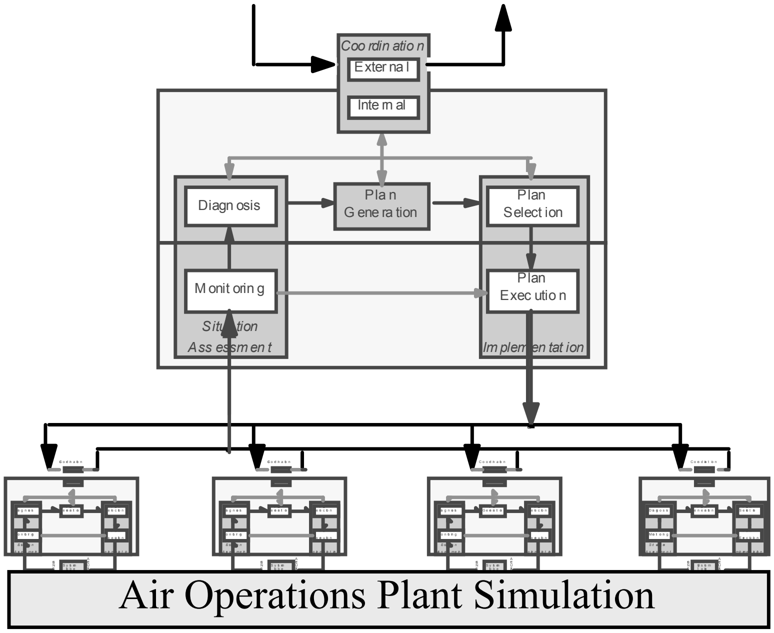


Figure 4.2- 2. Hierarchical View: Aggregated Plant for Master Level.

The closed-loop ***optimal control*** for each of the distributed elements is formulated as a receding horizon optimal control problem with the following attributes:

- ***Optimization:*** a time-varying "set point" control (e.g., a plan) and associated expected time history of system state values that would evolve if the plan were pursued are determined over a finite time horizon to optimize an objective function representing the desired system performance (in our case, the Commander's Intent).
- ***Control***: a high rate "perturbation controller" will augment the set point commands to stabilize the operation of the system and to ensure that the state of the system tracks the trajectory associated with the set point in the presence of disturbances.

Our approach achieves two objectives:

1. Provide a structure/framework for solving the air operations problem that resolves both resource conflicts across lower levels, as well as allocates objectives to the lower levels.

   This is done in a way that results in a degree of autonomy on the part of the lower levels in solving their decoupled problems. There are human-in-the-loop considerations on how one maps the decomposed problem onto the human organizational elements that must be ultimately responsible for planning and execution. Our approach defines the type of negotiation (e.g., iteration) and associated information exchanges among the levels required to arrive at a good, overall solution.

2. Use the data/information exchanges among problem-solving elements prescribed by the decomposition to form the basis of the real-time feedback when we "close-the-loop" on the decision-making.

   That is, in accommodating our inability to exactly model/formulate the problem due to the myriad of uncertainties and unknowns that prevail in a real warfighting situation, the sensitivity to those uncertainties is reduced using feedback (the purpose of closing the loop in even the simplest of control systems). Thus, the solution will evolve over time in response to the sensed state as well as new objectives provided by command levels.

Benefits: The decomposition via the theory of large-scale optimization ensures that enterprise-wide objectives are pursued, and enterprise-wide constraints are honored by every element of the hierarchical and distributed system. The control architecture ensures that feedback is employed to reduce the sensitivity of the system to disturbances, time delays, and model uncertainties.

4.2.2 Technical Challenges: Decomposition

The plan generation process must be coordinated across adjacent levels of the command and control hierarchy in order to meet enterprise-wide objectives, and there must be an internal control of the interactions among levels in order to ensure the stability of the overall plan generation process. The following basic questions must be answered in developing decompositions for large-scale enterprise operations:

- How many levels are required?
- How should problem solving be partitioned across levels?
- What constraints and objectives should be passed from level to level?
- What is the nature of the status that is passed from subordinate to superior levels?
- How is problem-solving best accomplished across levels?
- What happens when a level cannot meet its objectives and/or honor its constraints?
- To what extent should the decomposition reflect human-system interaction concerns?
- How might one develop a system wherein levels are established dynamically?

4.3 Closed-Loop Controller

An open-loop controller, no matter how accurate the internal models of the plant are, will slowly but surely begin making plans and issuing commands that take the plant into undesired states. Because random disturbances in the plant are constant, no controller can ever achieve the level of model resolution that reliably predicts the state of the plant at future times. For this reason, frequent feedback of the actual state of the plant to the controller is required. Providing this "toe-hold" on reality allows the modeling within the controller to be less than exact, but still sufficient for good overall performance. This assertion will be adequately demonstrated in Section 8, where we discuss our experimental results.

State information is fed back to each subcontroller, at each level in the control hierarchy, in essentially the same way. In fact, described in the next section is a template that we developed as a building block for piecing together more capable controllers.

4.3.1 Feedback Controller Template

The functional template for each of the elements within a command and control system is shown in Figure 4.3-1. This is a simplified version of Figure 4.2-1; note that the Plan Generation and Plan Selection functions of Figure 4.2-1 are now collectively known as the Planner function. Also, the terms *command* and *control nodes* and *elements* are used interchangeably in the remainder of the document. The four functions for intelligently carrying out given objectives for a system subject to various disturbances are shown, as well as the basic relationships they have to one another. Simplified operation of the template is straightforward and follows the basic observe-orient-decide-act sequence of an intelligent actor. In turn:

1. The *Monitor* accepts status from plant, estimates the state of the actual plan and world, compares actual state with expectations, and provides the Diagnoser with the state differences.
2. The *Diagnoser* accepts state differences from the Monitor, determines if replanning is warranted, and creates new problem for the Planner if it is.
3. The *Planner* accepts problems for solution from either an upper level controller or its own Diagnoser, splices a new plan with old plan, and provides the Executor with the new plan.
4. The *Executor* converts the plan representation to a form that can be executed by various actuation elements in the real world. These elements may include hardware, software, human, elements. The command plan is then sent.

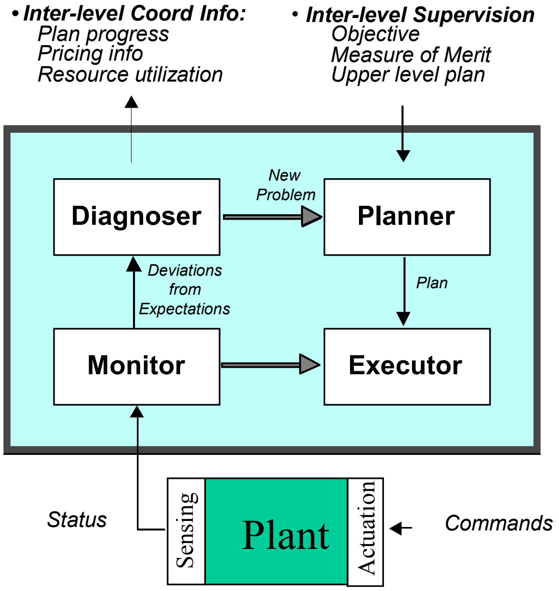


Figure 4.3-1. Functional Decomposition of a Command and Control Element.

The span and nature of the feedback needed for a particular controller depends on the scope and resolution of the plans being produced, and on how system is being modeled within the planner—its resolution, parameters, and so on. Some system disturbances may be too highly resolved (i.e., small) to be of interest to the controller. On the other hand, the resolution of some may be too low. Thus, the four components of a controller must be designed as a matched set in order for them to work properly.

4.3.2 Hierarchical Feedback

The controller template can be used to build up more capable and complex structures in different ways. One common structure, of course, is a hierarchy that mirrors the decomposed problem/system. An example representative of the JFACC air campaign problem is shown in Figure 4.3-2, where the three levels correspond to resource/target allocation for the theater, mission generation for aircraft at an air base, and routing for individual aircraft.

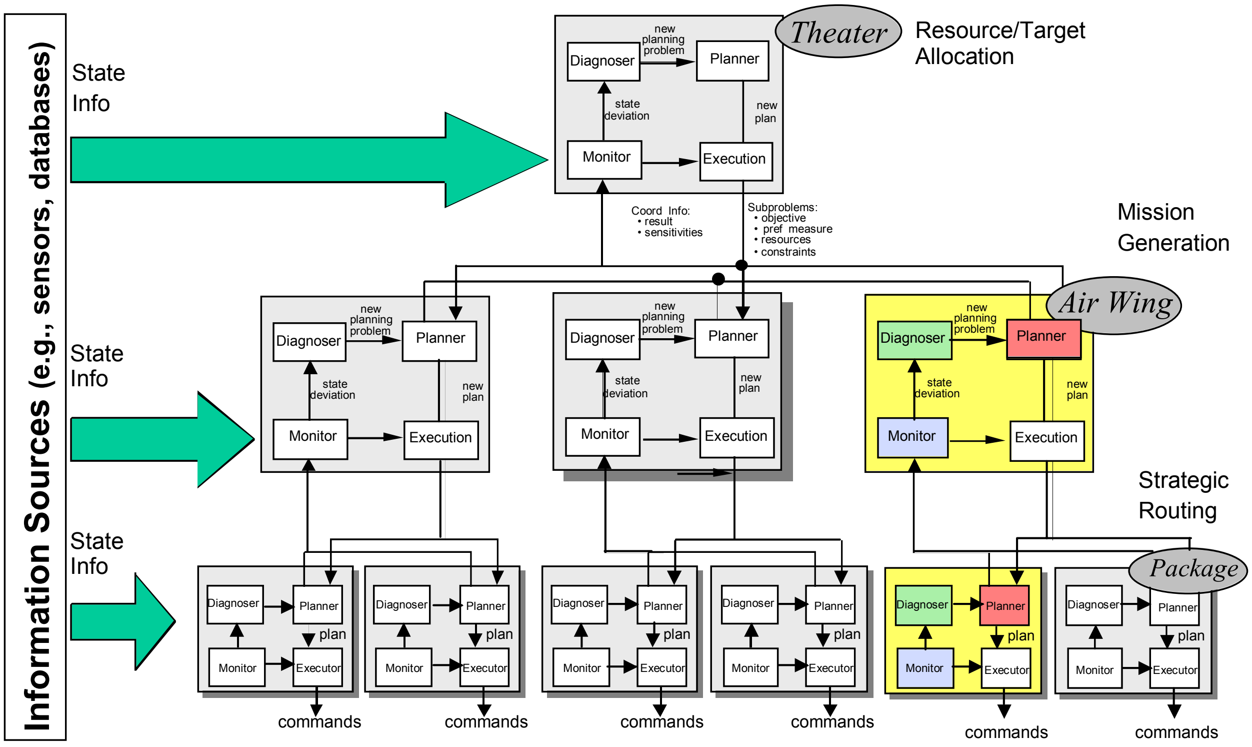


Figure 4.3-2. State Feedback in a Hierarchical Controller.

State feedback to a hierarchy of controllers, as shown, goes directly to each subcontroller at all levels. This feedback eventually precipitates replannings at various nodes in the hierarchy, which, in turn, may propagate replanning activity to either higher or lower levels. As noted in the previous section, the monitor in each subcontroller is tuned to "see" only changes in state information that its companion planner can use in its decision making. Higher level monitors will tend to see only big changes; lower monitors will tend to see only small changes.

State information can generally stream to the hierarchy in a couple of different ways:

1. ***Synchronous:*** In this case, all monitors in the hierarchy are updated simultaneously, periodically, or aperiodically, although different levels may updated at different rates. Each companion Diagnoser determines whether or not to replan. With synchronous updates, it may happen that a number of subcontrollers throughout the hierarchy will want to replan at the same time. When computational resources are limited, as may be the case if the entire hierarchy is hosted on a single central processing unit (CPU), it may be advisable to sequence these replannings according to some reasonable criterion. Obviously, controllers higher in the hierarchy would be given precedence over lower level controllers, because their actions may indeed precipitate lower replannings anyway. Less obvious are cases where two

or more superior controllers at the same level need replanning. On the other hand, if controllers are distributed and have their own computational resources, no harm is done if they are allowed to replan at will.

2. ***Asynchronous*:** Here, independent streams flow state information to each monitor periodically or aperiodically and at rates that vary dramatically over the hierarchy. The probability that two or more subcontrollers will want to replan at the same time is quite small, although clusters of replanning events are likely. Again, if computational resources are limited, it may be worth devising schemes that prioritize the sequencing of these clustered replanning events, or employ techniques where replannings can be interrupted and scraped, in favor of a higher level event.

4.4 Multilevel Optimization

Analytical approaches to decomposing *large-scale optimization problems* and the essential role played by subproblem coordination (which is itself formulated as an optimization problem) have been developed over the last three decades.[1] The formal analytical developments help to establish methodologies for achieving decompositions wherein the subproblems are properly coordinated via a higher *Master* level. These approaches have been extended in the development of methodologies for decomposing *large-scale control problems* for linear dynamic systems and quadratic cost functions.[2] The discussion below of multilevel decompositions is intended to be qualitative. Technical details and conditions can be found in the references cited above.

Consider the typical problem statement:

$$\min_{\underline{x},\underline{y}} f(\underline{x}, \underline{y})$$

$$\text{subject to} \qquad \underline{g}(\underline{x},\underline{y}) \leq \underline{0}$$

$$\text{where} \qquad \underline{x} = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \underline{x}_N \end{bmatrix}$$

The vector $\underline{x}$ is shown to be composed of subvectors $\underline{x}_i$, which will later be associated with the N subproblems at a lower, subordinate level of the decomposition. The vector $\underline{y}$ corresponds to the variables that couple the subproblems in either the objective function f and the constraint vector $\underline{g}$ or both. The problem can be rewritten in terms of a Lagrangian L with Kuhn-Tucker multiplier vector $\underline{\gamma}$

$$L(\underline{x},\underline{y}) = f(\underline{x},\underline{y}) + \underline{\gamma}^T \underline{g}(\underline{x},\underline{y})$$

The decomposition of the optimization problem is achieved by creating a superior or master-level problem whose solution produces the value of the coupling vector $\underline{y}$ or the multiplier $\underline{\gamma}$. The former is referred to as "interaction prediction" and the latter as "goal coordination" or "price coordination." To establish the decomposition and define the superior and subordinate problems, the Lagrangian is rewritten as a sum of decoupled Lagrangians (i.e., setting a value for $\underline{y}$ leads to separability of both the objective function f and the constraint vector $\underline{g}$)

$$\begin{aligned} L(\underline{x};\underline{y}) &= \sum_{i=1}^{N} [f_i(\underline{x}_i;\underline{y}) + \underline{\gamma}_i^T \underline{g}_i(\underline{x}_i;\underline{y})] \\ &= \sum_{i=1}^{N} L_i(\underline{x}_i;\underline{y}) \end{aligned}$$

In this case, given values for $\underline{y}$, each of the N subordinate levels is responsible for solving a decoupled optimization problem associated with one of the decoupled Lagrangians, L_i

$$\min_{x_i} f_i\,(\underline{x}_i\,;\underline{y})$$

$$\underline{g}_i\,(\underline{x}_i;\underline{y}) \leq \underline{0}$$

Iterations between the upper and subordinate levels ***are required to achieve an optimal solution***. The nature of the iterations is a direct byproduct of the decomposition.

The variables $\underline{y}$ might represent coordination points or times that couple adjacent mission phases, where the ith subproblem would be the planning problem for the ith mission phase. For a mission involving multiple strike packages, these variables might be parameters that enforce coordination across those missions. In that case, the ith problem would be the mission for the ith strike package. Of course, those missions would be further decomposed for the individual aircraft composing the strike package.

The formal details of the decomposition for the price coordination method depend on the nature of both *f* and $\underline{g}$. For some problems, price coordination alone may be insufficient to completely decouple the problem. In general, the objective is to achieve unconstrained subordinate level problems of the form

$$\min_{x_i} f_i\,(\underline{x}_i) + \underline{\gamma}_i^T\,\underline{g}_i\,(\underline{x}_i)$$

where the nature of the problem has allowed us to avoid dependence on the coupling variables $\underline{y}$. The spirit of price coordination is that the superior or master level does not explicitly set values for the coupling variables, but sets "prices" or penalties (i.e., values for the multipliers) for violating the constraints. As with the interaction prediction approach, iterations between the upper and subordinate levels are required to achieve an optimal solution.

The price coordination and interaction prediction approaches can be combined ("mixed") wherein the superior or master-level fixes values for any subset of the coupling variables that is sufficient to decouple the Lagrangians, and sets prices for constraint violations that are evaluated at the fixed values of the coupling variable $\underline{y}$.

4.5 Modeling and Simulation

4.5.1 Modeling Rationale

Both the mathematical models and the software implementation of the Air Operations Controller are sufficiently complex so that the only feasible approach to support development and testing is numerical simulation. The physical elements that are modeled include aircraft, air bases, ground-based air defense threats, weapons, and targets. Environmental elements include weather and physical features that affect the successful acquisition of time-critical targets by strike aircraft. Information elements include the state of objects, including locations, fuel status, damage status, as well as commands that determine intended future actions. The execution of commands is simulated with commanded actions modulated or negated by physical constraints such as proximity, fuel state, state of ordnance stores, stochastic factors, etc. For example, a commanded weapons release is not executed if the simulated aircraft has not arrived within proximity of the prescribed release point or is not carrying the prescribed weapon type. Similarly, commanded aircraft recovery will not occur if not within proximity of an operating base, commanded refueling will not occur unless within proximity of the prescribed tanker location, and aircraft that run out of fuel because of planning deficiencies will be reported as lost.

Potential human decision elements include the decisions and actions of JFACC planners, pilots, and air defense system operators. The actions of JFACC planners are abstracted out across the interface that defines the experiment and scenario. In other words, the parameters and controls on the JFACC Air Operations Controller are initialized, and any changes during an experiment are scripted. The JFACC Air Operations Controller operates in a fully autonomous mode during the experiment, with perturbations representing human interactions either scripted or represented by stochastic models using pseudorandom numbers. This is necessary to allow the replication of experimental conditions and to provide other aspects of a systematic experimentation environment.

Human factors experimentation regarding the interaction of Air Operations staff with the JFACC Air Operations Controller is not within project scope.

Pilots are assumed to properly execute commanded mission tasks and the enforcement of physical constraints and proximity tests are assumed to parallel the actions of human pilots. Simulation resolution does not encompass detailed tactics or other detailed actions mediated by human judgement. Pilot performance in target acquisition is modeled stochastically. There is no simulation of pilot-based replanning or reaction to contingencies. The decisions of air defense system operators is folded into the overall performance model of air defense threats, with a stochastic model for the likelihood of engagement given a physical condition of engageability.

The overall technical approach to Air Operations Simulation is to apply the lowest fidelity and least complex model that will represent the desired behaviors and capture the desired interactions with the JFACC Air Operations Controller. In this vein, aircraft dynamics are represented by great circle, constant speed propagation between commanded waypoints in place of integration of dynamical equations of motion. Details such as take-off roll and acceleration to cruise speed are unnecessary to simulate explicitly since the effect on mission execution timeline is either insignificant or is easily modeled otherwise. The advantages of this approach are manifold:

- It is unnecessary to specify and validate hundreds or thousands of parameters that are required by the detailed models.
- It is unnecessary to validate the many components of the detailed models in the many modes and configurations that will be sampled in the overall air campaign.
- It is unnecessary to incur the computation time penalty for solving hundreds of differential equations.

The principal models are:

- Time duration for activities, such as ground preparation, aerial refueling, target acquisition and weapon release, aircraft recovery.
- Great circle constant speed propagation between commanded waypoints, with banked-turn loitering on waypoint arrival prior to the specified time of arrival.
- Fuel use and remaining unrefueled endurance that is linear with time.
- Target fractional damage that is deterministic according to weaponeering specifications.
- Stochastic air defense threat interactions.

Each aircraft is separately tasked, with mission package coordination handled solely by the Air Operations Controller. Mission packages often but not always originate from the same base, with provision for dynamic retasking and dynamic package formation at assembly points. There is no "Commander logic" or rule-based system for aircraft autonomously determining mission aborts and initiating return to base behavior. Absent any error in the mission tasking, there is no contingency in the Air Operations Simulation where this would be required. Threat interactions result in either no change or a shoot-down of mission aircraft. Aerial refueling is explicitly tasked. The Air Operations Controller includes a significant safety margin of flight endurance to mitigate errors in fuel use models.

4.5.2 Objects, Processes, and Control Parameters

The purpose of the Air Operations Simulation is to:

- Provide machinery for defining the experiment to be executed, including the scenario, control actions, discrepancies between truth and controller models, and Monte Carlo samples.

- Propagate the state of the "Air Operations world" and transmit selected state information to the "Air Operations Controller."
- Receive command information from the "Air Operations Controller" and propagate the world model until the next controller interaction.
- Provide machinery for monitoring, analyzing, and generating archived results.

The types of information that are reported to the Air Operations Controller include the following:

- Simulated time.
- Aircraft instances.
- Target instances.
- Aircraft configurations (standard conventional loads).
- Mission package configurations.
- Weapon types.
- Air bases.
- Tanker orbit locations.
- Commander’s Intent tables.
- Threat tables and parameters.
- Logistical cost tables.

The Air Operations Simulation propagates aircraft and target states according to tasked mission activities and according to interaction models with stochastic elements. To provide additional scenario flexibility, the simulation also propagates externally scripted objects that are not explicitly controlled by or reported to the Air Operations Controller.

The processes that are modeled include:

- Aircraft ground preparation into a commanded mission configuration (standard conventional load), including fueling, arming, mission planning, pilot assignment, and maintenance between missions, culminating in mission launch.
- Physical propagation via commanded waypoints with loitering for time adjustments and depletion of remaining fuel endurance.
- Assembly point rendezvous.
- Aerial refueling.
- Stochastic target acquisition of time-critical targets.
- Weapon release with target damage state propagation.
- Aircraft recovery (landing).
- Ground-based air defense interactions with possible aircraft attrition.
- Target reconstitution.

Discrepancies between the "truth state world model" and the information sent to the "Air Operations Controller" include:

- Threat density errors.
- Weaponeered weapon effectiveness errors.

Dynamic scenario perturbations include:

- Unanticipated changes in campaign phase time transitions.

- Temporary base closures.

Additional scenario variations are generated by specifying different random number strings for Monte Carlo sampling.

Controls that affect the Air Operations Controller include:

- Controller cycle times and event thresholds for event-driven replanning.
- Planning time horizon.
- Decomposition and negotiation parameters.
- Retasking cost.

UCAV-type scenarios can be specified by altering both the maximum mission risk specification as well as the retasking cost for UCAV-type aircraft.

Processes that are not modeled *explicitly* include:

- Intelligence, surveillance, and reconnaissance leading to target development.
- Bomb damage assessment.
- Relocation of ground targets.
- Attrition and relocation of ground-based air defenses.
- Counter-air activities.
- Tasking of aircraft not contained in strike packages.

Targets are entered into the JFACC Air Operations Controller system at scripted times with approximately half of the total number of targets available at commencement of operations and the other half discovered at randomly sampled scripted times over the course of several days of operations. Bomb damage assessment is assumed to be perfect and without delay.

For the same reasons stated earlier with regard to model fidelity, it is advantageous to avoid explicit modeling of processes when simpler, implicit models will suffice for controller development and testing. Only those aspects of real-world fidelity that relate to essential controller behavior and interactions have been included in the current JFACC Air Operations Simulation.

4.5.3 Simulation Structure

All simulations can be categorized as event-based, time-stepped, or hybrid. In actual practice, event-based simulation structures need to incorporate time-stepping for numerical integration of some continuous-time processes, and time-stepped simulation structures incorporate event-forecasting and sorting for efficient time-step control. Although both approaches are workable for this application, the time-stepped approach has been adopted for the JFACC Air Operations Simulation. It is believed that this approach provides greater physical insight into interaction processes leading to more rapid debugging during the development of both simulation and controller. The time-stepped simulation method is adequately efficient, as evidenced by the fact that only a small percentage of the overall clock time for an experiment is consumed by the simulation. For the 80-aircraft scenario, the JFACC Air Operations Simulation typically completes a cycle of 4 simulated hours at 30-s increments in about 5 s of wall clock time, and then waits several minutes for the receipt of commands from the JFACC Air Operations Controller.

An essential feature of the experimental ground rules is the simplification that the time-to-plan is not a feature of the experiment. In other words, the simulation and the controller run in consecutive, cyclic fashion rather than concurrently, and simulated time is suspended during the execution of the JFACC Air Operations Controller. In a go-to-war implementation, the time-to-plan would be a significant issue and it is, of course, recorded during the experiments. Part of the rationale for this simplification is a significant reduction in implementation complexity, an emphasis on early development of loop-closure control theory applications to the Air Operations domain, and a belief that real-time issues to be addressed in any go-to-war implementation would be mediated by the advance in processor technology in the interim. Hence, anticipated requirements for time-to-plan were relaxed by a factor of a few in the selection and development of controller algorithms.

The general problem to be addressed in the JFACC Air Operations Simulation is the simultaneous propagation of the state of a number of interacting objects. The dynamics of object propagation depend on the world model environment and on the state of other objects. In order to numerically decouple the interactions between objects, a quasistatic approximation is applied that enables independent calculation of object dynamics, assuming that the state of the other objects with which it interacts is static during the propagation interval. If objects are closely coupled by mutual interactions, as in the example of aerial combat, then the propagation interval would of necessity be very small in order validate the approximation. When the characteristic time constants of interacting objects are widely separated, as in the case of a set of objects with really fast dynamics interacting with a set with really slow dynamics, the time steps appropriate for the objects with fast dynamics will intrinsically validate the quasistatic property for the objects with slow dynamics.

Given that the JFACC Air Operations Simulation scope does not include air-to-air engagements, the primary dynamics at play are the motions of strike packages. The motions of targets and threat systems on the ground are not modeled explicitly. The likelihood of acquisition of fleeting ground targets is stochastically modeled. The dynamics of offensive weapons are not modeled explicitly; on completion of a defined duration for the weapon release activity, the target damage state is altered. Finally, explicit modeling of the dynamics of air defense threat interaction is also avoided. If a strike aircraft is engaged by ground defenses, a stochastic model determines the outcome, and the strike aircraft is either shot down or continues its mission. Since the fidelity of the aircraft propagation model is appropriate to the analysis of extended air campaigns, it would make little sense to incorporate explicit play of countermeasures and jinking maneuvers during the minute or two of active engagement. Each of these interaction events could indeed be analyzed at a high level of fidelity, but the JFACC Air Operations Simulation only requires the results of this analysis as translated into a few parameters for use in stochastic models.

Most of the dynamics at play in the JFACC Air Operations Simulation relate to the duration of events whose outcomes are either deterministic or are sampled stochastically. The physical motion of aircraft is modeled as constant speed great circle paths between commanded waypoints. The activity of readying an aircraft for a mission is represented as a time delay with a serviced, fueled, and armed aircraft taking off at the conclusion of the delay. The activity of rendezvous and aerial refueling is modeled as a delay with activity completion after the specified nominal duration. The activity of terminal area weapon release and the activity of aircraft recovery at base are similarly modeled as events of finite and specified nominal duration without explicit modeling of the details. Finally, only the outcome of the ISR process that results in target development is modeled by time-scripted introduction of targets. Because events for different objects have arbitrary time phasing between respective events, the overall state propagation scheme employs an outer time loop or frame time at which all object propagations are synchronized. In Figure 4.5-1, time advances to the right and the time steps for different objects are depicted as the interval between lines on horizontal bars representing the timeline for each of four objects. The time steps may be different for each object, and variable depending on the event or mode of an object. Synchronization between objects at specified "Frame Times" is achieved by shimming the time steps for each object to coincide with the "Frame Time" boundaries.

The overall simulation logic is described in pseudocode by an outer loop on frame time:

```
DO WHILE (eTime .lt. etStopTime)
  eTime = eTime + dTframe
  call RunOneFrame( )
END DO
```

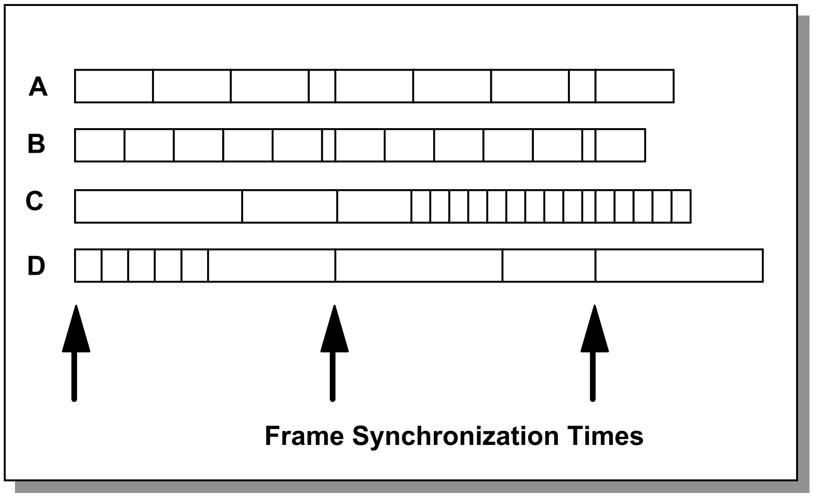


Figure 4.5-1. Illustration of Multiple Time Scale Synchronization.

and an inner loop over all objects in procedure RunOneFrame( ):

```
     SUBROUTINE RunOneFrame( )
       .
       .
          DO WHILE (eTime .lt. etStopTime)
C  Test for exit conditions
       eTime = eTime + dTframe
C  Loop over all objects
      DO (theObj=1,NumObjs)
       DO WHILE (Tme_curObj(theObj) .lt. eTime)
     theActivity = current_activity(theObj)
               select case (whichactivity)
                         implement the current activity
               end select
       END DO
     END DO
```

If an activity completes within the prescribed time step, the next activity in the list of commanded activities for that object is invoked.

A number of details have been elided for clarity in the pseudocode example, including:

- Initialization.
- Checking for exit conditions.
- Checking for proximity of escorts.
- Implementation of loiter and shootdown functions.

- Reading of command files and writing of state files.
- Logging runtime reports.
- Generating runtime graphical displays.
- Introducing perturbations.

Instead of forcing all processes and objects to apply the smallest time step required by any object, the multiple and variable step size structure enables the time-stepped simulation method to efficiently propagate a complex system without significant run-time penalties. The design and implementation of the JFACC Air Operations Simulation has been highly successful in providing timely support for the development of and experiments with the JFACC Air Operations Controller.

4.6 Interface Design

The JFACC Air Operations Simulation and the JFACC Air Operations Controller are two separate processes, implemented in different computer languages (FORTRAN and C++) and hosted on different computers running different operating systems (NT and Linux). This mechanization allowed for the separate development of these components, and precluded any inadvertent sharing of world state information that was not defined explicitly on the interface between the two components.

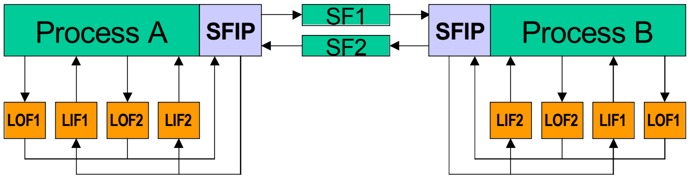
A number of possibilities exist for implementing interprocess interface, given separate host machines for the two processes. These include high-level architecture (HLA), common object request broker architecture (CORBA), sockets, and file-based communications. Anticipating that computer science complexities in the first three approaches will require extensive debugging with dumping of interface data to file for off-line examination, the latter approach is to skip the computer science tribulation and go directly to file-based communication. This is entirely feasible since the interaction between processes is cyclic, with many seconds or minutes between distinct communications. The entire transcript of "state files" written by the simulation and read by the controller and the "command files" written by the controller and read by the simulation are also saved to a "log file." The log file has been indispensable for analyzing and recreating snippets of puzzling behavior as well as overall debugging. It also provides a "replay mode" for the simulation, wherein an experiment can be replayed in its entirety with a live simulation, but "offline" from the controller. This capability also enables simpler more effective debugging procedures.

The low-technology lingua franca between different processes and computer languages is ASCII file I/O and local area network access to a common set of shared files. Duplex communication between two processes requires two shared files. In actual implementation, the semaphore logic to pause either process while the other is generating results requires a set of acknowledgment messages that are most reliably implemented via an additional set of shared ASCII files. Either process can be started first, and both look for the presence of files generated by the other process during the automatic synchronization that is accomplished at the beginning of an experiment.

Figure 4.6-1 illustrates the mechanization of the shared file interface concept.

The information on the shared files is encoded with data tags to facilitate automated parsing in a customized XML-like schema. Textual descriptors are also appended to each data line to facilitate human interpretation during debugging. An example of the format is given below:

```
[Airbase]
id                         Gaan
latitude                   -4.6667
longitude                  144.9667
altitude                   50.0000
beginAvailability          0
endAvailability            1000000
```



SFIP	= Shared File Interface Procedure (Fortran/ C)
SF1,2	= Shared Files between processes
LOF1,2	= Local output file from process
LIF1,2	= Local input file from process

- All interprocess messaging is through SFIP (symmetric to all processes)
- Local files contain message formatting (e.g., tags)
  - Use as many local files as required (illustration depicts using 4 files)
- SFIP invoked to:
  - Set up connection and begin synchronization of processes
  - Send a specified LOF to a SF and wait for short acknowledgement
  - Receive any message from a SF and send an acknowledgement

Figure 4.6-1. Shared File Interprocess Concept.

The bracketed [Air base] tag is used by the parser to select functions that positionally read the items expected for air base objects. The textual descriptors are ignored by the parser and are only provided for human readability. Hierarchical data are represented by nested tags. The interface is described in more detail in Section 7.

4.7 Scenarios and Experimentation

The scenario context and some details were developed by the DARPA JFACC Program Office. To provide a real geography, the operational area was set in the vicinity of New Guinea, with the Indonesian province of Irian Jaya on the western half of the island being identified as the fictitious aggressor state of "West Cyberland" and the eastern half of the island, comprising most of Papua New Guinea, being identified as the allied state of "East Cyberland." A map of the area is shown in Figure 4.7-1.

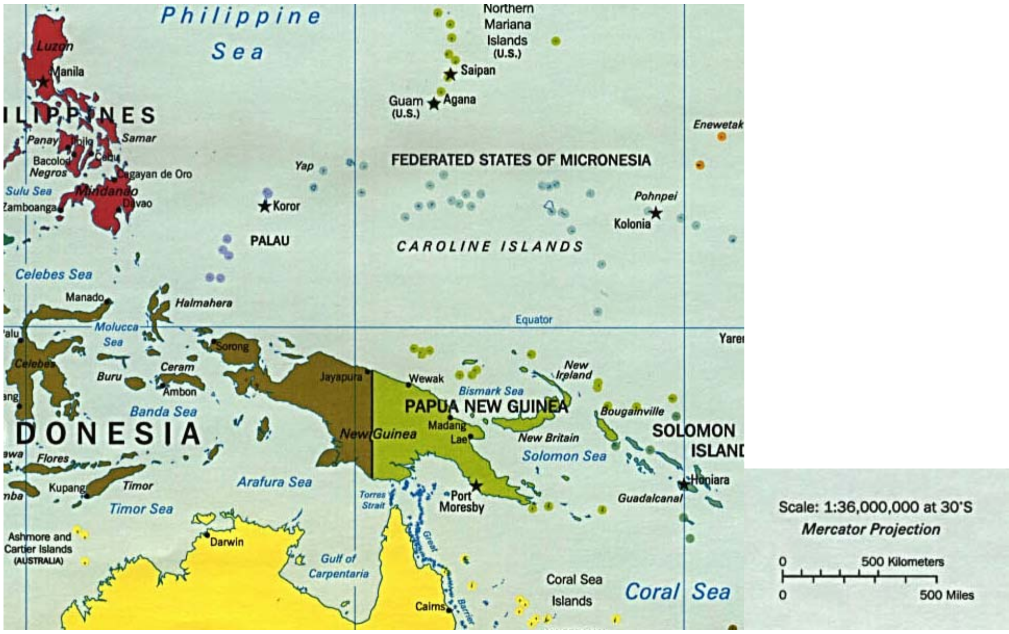


Figure 4.7-1. Notional Operational Area.

The important geographic features are that the air operations targets are primarily in West Cyberland, with some an invasion salient penetrating into the northwest corner of East Cyberland; the "in-country" bases are in East Cyberland; there are bases that can be used in Guam and Darwin, Australia, and one-way flight distances within the island can range up to 1500 km. The large scale of the operational area implies that tactical aircraft operating from East Cyberland bases will frequently require one or several aerial refueling activities to complete missions against West Cyberland.

The elements of the scenario with which we are most concerned include:

- Base locations.
- Numbers and types of deployed aircraft.
- Target locations and associated weaponeering.
- Air defense threat laydown.
- Campaign time and objectives.

The original guidance from the Program Office allowed the numbers of deployed aircraft, the total number of targets, and some other elements to be free parameters from the perspective of defining meaningful experiments. The descriptive timeline indicated a duration for the air campaign of about 1 week. Additional elements that need to be defined include aerial refueling locations and assembly point locations.

Given the geography, the top-level parameter is the number of strike and supporting escort aircraft vs the total number of targets to be addressed. The Draper scenario defined a deployment of 80 strike-package aircraft (attack aircraft, high-speed antiradiation missile (HARM)-shooters and jammers) amounting to the better part of an Air Expeditionary Wing, 3 refueling tanker locations, and a nominal case of 313 targets with an excursion of 910 targets. The former case represents a case where a little over 3 days of high-tempo campaign time were required to reduce the target set whereas the 910 target case represented a situation where "service rate capacity" was stressed more significantly. In both cases, approximately 261 air defense threat systems were distributed, principally around targets and access routes to targets. The air defense threats were distributed as shown in Figure 4.7-2 where the grid box resolution is 30 km.

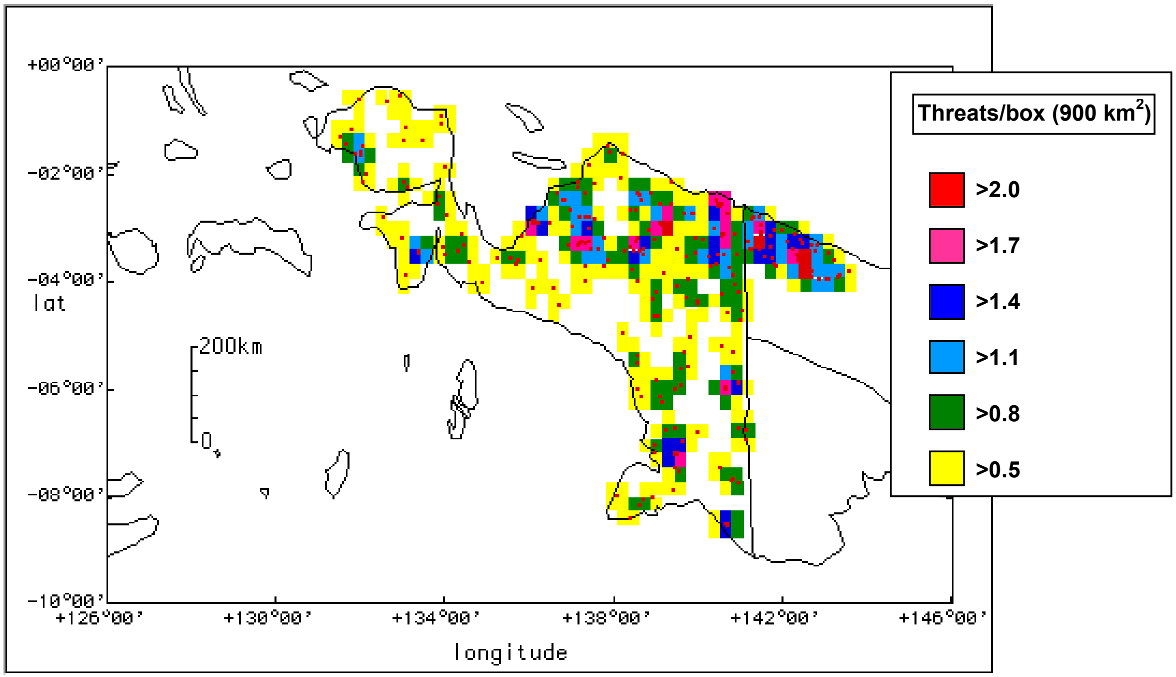


Figure 4.7-2. Distribution of Air Defense Threats.

The theme in constructing the details of the scenario were to minimize complexity that was unnecessary to the development and assessment of controller performance or that impeded interpretation of closed-loop behaviors. For example, different payoff functions were applied to different target classes according to what might be expected for those classes of target systems. The nonlinear payoffs, however, introduced difficulty in interpreting decisions about which targets were selected for prosecution and when. Hence, the decision was made to perform all experimentation with linear payoff functions. Some features were judged to be important, including the definition of time-critical targets for 10-15% of all targets, and Commander's Intent inputs changing across campaign phase boundaries at 2 days and 4 days into the campaign.

The overall experimental structure included the following variations:

- No threats vs normal threats.
- 24-h cycle vs 4-h controller cycles.
- 313 vs 910 targets.
- No time-critical targets, a nominal number of time-critical targets, and all targets time-critical.
- Nominal risk vs no risk aversion.
- Nominal retasking cost vs no retasking cost.
- Monte Carlo samples with different random numbers for statistical analysis.
- No weaponeering mismodeling perturbations vs some weaponeering mismodeling perturbations.
- No threat mismodeling perturbations vs some threat mismodeling perturbations.
- No changes vs unanticipated changes to campaign phase time perturbations.
- No perturbations vs temporary base closure perturbations.
- Decomposed and undecomposed tasking.
- Heuristic vs optimal methods.
- Heuristic, with and without negotiations.
- All-UCAV cases, i.e., with no risk or retasking aversion.
- Nominal cyclic vs event-based controller architecture.

Not all combinations were generated: most of the cases were executed with 313 targets, nominal numbers of time-critical targets, nominal threats, heuristic, decomposed, and non-negotiating controller. After establishing the statistical validity of experimental results, the technical approach usually consisted of contrasting a reference case with an excursion from that case. The 24-h cycle case was considered the baseline for the current state of Air Operations response time, and the comparison with shorter cycle times under varying conditions and perturbations were particularly key experiments. The ability to maintain campaign performance while replanning to adapt to significant perturbations was another key result. For the 910 target case, the controller generated 374,000 activities in 7 days of strike operations with 4-h replanning cycles. Of the 1600 sorties generated, approximately 900 were reassigned, mostly during ground preparation. Although system capacity was reduced by the base closure perturbation, the controller maintained a high level of performance in the face of massive reorientation and regeneration of tasking in response to this perturbation.

5 Controller Design and Implementation

5.1 Plan Generation Models and Algorithms

This section describes the specifics of the plan generation models and algorithms as applied to the strike planning problem, based on the theory underlying our approach to optimization-based plan generation.

5.1.1 Objective Function Weights from Commander's Intent

As described in Section 4.1, the Commander's guidance is concretely expressed as a table of weightings with rows varying over target functional categories and columns expressing geographic regions or groupings. There may be a number of tables expressing guidance inputs for each campaign phase. Given finite strike resources, the table expresses the relative value to the Commander of achieving damage to targets in different slots. Instead of a rigid, top-down allocation of resources, this table is used to establish objective value for alternative strike plans. Targets in lower weighted slots may, in fact, be incrementally favored over other targets because of collective effects.

The payoff function is applied separately in each category/region cell and may be linear for targets with independent valuation or nonlinear for targets with collective valuation. The overall or aggregate value of all targets is formulated as the weighted sum of payoffs for each slot with weightings specified by the Commander's guidance tables. It should be noted that the payoff metric can be evaluated with the current damage state of targets (i.e., strikes already executed) or with the projected damage state for current plans (i.e., including future planned strikes). The former is useful for monitoring actual target value accrued as a function of time as opposed to projected accruals, including the effects of future planned strikes. Of course, the actual damage state achieved may be different than the projected damage state for any strike by virtue of simulated discrepancies in weaponeering estimation.

5.1.2 General Problem Formulation

The overall objective of the air strike operations planning problem to be solved by our controller is to maximize the target value achieved over a specified planning horizon subject to the constraints of the air operations environment and the available resources. This section gives the general mathematical formulation of the problem being solved.

Objective: select from among the available targets (**T**), the available aircraft packages (**P**), the feasible routes (**R**) for the aircraft composing those packages, and the feasible weapon configurations (**W**) for those aircraft that maximize the total expected return as defined by

$$\underset{\substack{P \in \mathbf{P} \\ R \in \mathbf{R} \\ W \in \mathbf{W} \\ T \in \mathbf{T}}}{Max} \sum_{tar_i \in T} \left[E(v(tar_i)) - E(\cos t(tar_i)) \right]$$

where

P = the set of selected aircraft packages from all available packages **P**

R = the set of routes chosen for the aircraft in P from the fuel-feasible routes **R**

W = the weapon configurations (weaponeering) chosen for each of the aircraft in P from among the feasible configurations **W**

T = the set of targets selected from the available set **T**

tar_i = the i^{th} target in the target set T

$v(tar_i)$ = the value to the campaign accrued for target tar_i when the desired level of damage is achieved

E(v(tar$_i$)) = the expected value of target *tar$_i$* where the expectation is taken over the uncertainty in reaching, finding, and successfully damaging the target to the specified level. This expectation is a function of the air defense threat, the selected package (e.g., level of escort), the selected routes for the aircraft within the package, the selected weaponeering for the package, and the time sensitivity (e.g., mobility) of the target, and the time that the routes will be executed

E(cost(tar$_i$)) = is the expected cost of prosecuting target *tar$_i$* where the expectation is taken over the attrition cost of attacking the target and is a function of the chosen routes, aircraft package (e.g., escort level), and air defense threat laydown. Also included in the cost are aircraft flight hour costs and fuel cost, both of which are modeled as deterministic

Constraints: The constraints for the problem relate to the availability of aircraft and weapons, performance of aircraft and weapons, and the locations of tankers available for refueling:

- Aircraft and weapons resource supply.
  - Aircraft supply constrains the aircraft packages that can be assigned.
  - Weapon supply constrains the weaponeering that can be assigned.
- Aircraft and weapon performance.
  - Aircraft performance constrains the routes aircraft can fly, including the speed at which the aircraft can fly and distance they can travel without refueling.
  - Aircraft performance constrains the weapons aircraft can carry.
  - Aircraft performance constrains the effectiveness of using escorts.
  - Weapon performance constrains the effectiveness of weapons against individual targets.
- Tanker Locations.
  - We assume tanker locations are prespecified, which constrains routing and aircraft package to target assignments.

5.1.3 Integer Programming Formulation

An Integer Programming formulation of the problem defined in the previous section is presented below in Eqs. (2) through (6). The decision variables, x_{ijn}, are binary variables representing the assignment of strike packages from bases to targets. Each x_{ijn} is set to one if base *i* supplies strike package *n* to target *j*, otherwise, x_{ijn} is set to zero. Target-strike package preference information is represented by the coefficients $PREF_{jn}$ in the objective function (Eq. (2)). These coefficients reflect a combination of target valuation (per Commander's Intent) and the probability of successfully prosecuting target *j* with package n. The objective of this integer program is to maximize the sum of target preferences achieved through the assignment represented by the x_{ijn}; only if x_{ijn} = 1 is the value $PREF_{jn}$ added to the objective function.

$$\text{maximize}_x \quad \sum_i \sum_j \sum_n PREF_{jn} x_{ijn} \qquad (2)$$

$$\text{subject to} \quad \sum_i \sum_n x_{ijn} \le 1, \forall_j \qquad (3)$$

$$\sum_n \sum_j x_{ijn} QTY_{ajn} \le AVAIL_{ai}, \forall_a, i \qquad (4)$$

$$y_{ij} = \sum_n x_{ijn} \qquad (5)$$

$$x_{ijn} \in \{0,1\}, \forall_i, j, n \tag{6}$$

The constraints in Eq. (3) ensure that each target is hit by no more than one strike package. For target *t* to be hit by more than one strike package, more than one of the x_{ijn} would be one, therefore the sum over *i* and *n* of x_{ijn} would be greater than one.

The constraints in Eq. (4) prevent the math program from using more aircraft from a base than are available. The constants QTY_{ajn} represent the number of aircraft of type *a* required to hit target *j* using strike package *n*. Thus, the sum over all strike packages assigned from base *i* times the number of aircraft of type *a* used in each package must be less than the total number of aircraft of type *a* available from base *i*, represented by the constant $AVAIL_{ai}$.

Finally, the constraint in Eq. (5) serves to keep track of the targets that each base is being tasked to hit. Equation (5) forces variable y_{ij} to be one if aircraft from base *i* are striking target *j*, zero otherwise.

5.1.4 Decompositions

Below, we decompose the integer programming formulation defined above according to two of the approaches described earlier. The first decomposition is of the interaction prediction, or feasible type. The second decomposition is of the price-coordinated, or infeasible type. Note that these decompositions are not unique, and that alternative decompositions are possible.

We have chosen to develop a detailed solution for the interaction prediction decomposition as it most closely resembles the current practice and overlays well onto current military organizational structures. Consequently, it is the most likely decomposition to be accepted by the military for a near-term transition. In Section 5.2.4.2, we provide a detailed simplified example to illustrate the mechanics of this decomposition.

5.1.4.1 Interaction Prediction Decomposition of Math Programming Formulation

Consider a decomposition of the strike-planning problem in which an upper-level C^2 node assigns targets to bases with each base responsible for determining the strike packages and their associated routes to assign to each target. Such a decomposition is justified because the detailed planning required of a base to execute a strike is strongly influenced by its local resource constraints, and the information resulting from this detailed planning is not easily represented in data that can be readily communicated to or used by the upper level.

The decomposition comprises a master problem and a set of subproblems. For our decomposition, each subproblem is associated with a base, and the master problem is associated with the upper-level C^2 element. The base problems cannot be solved independently without coordination from the upper level because of constraint 2; namely, without coordination, multiple bases would likely strike the same target in the same wave, violating the constraint that each target be struck by no more than one strike package. The upper-level coordination addresses this issue by assigning nonoverlapping sets of targets to the bases.

To decompose the problem via interaction prediction, we introduce dummy variables σ_{ij} that are constrained to be equal to the y_{ij}. The master problem will set the σ_{ij}, which the subproblems will use to coordinate their solutions. The formulation is presented below, augmented with the σ_{ij} in Eq. (11). In effect, the σ_{ij} represent the assignment of targets *j* to base *i*.

$$\text{Maximize}_x \quad \sum_i \sum_j \sum_n PREF_{jn} x_{ijn} \tag{7}$$

$$\text{subject to} \quad \sum_i \sum_n x_{ijn} \le 1, \forall_j \tag{8}$$

$$\sum_n \sum_j x_{ijn} QTY_{ajn} \le AVAIL_{ai}, \forall ai \tag{9}$$

$$y_{ij} = \sum_n x_{ijn} \tag{10}$$

$$y_{ij} = \sigma_{ij}, \forall i, j \tag{11}$$

$$x_{ijn} \in \{0,1\}, \forall i, j, n \tag{12}$$

We can dualize the constraint on the σ_{ij} (Eq. (11)) by appending it to the objective function with Lagrange multipliers λ_{ij} to produce the following equivalent problem

minimize$_\lambda$ maximize$_x$ $$\sum_i \sum_j \sum_n PREF_{jn} x_{ijn} + \lambda_{ij}(\sigma_{ij} - y_{ij}) \tag{13}$$

subject to (14)

$$\sum_i \sum_n x_{ijn} \leq 1, \forall_j$$

$$\sum_n \sum_j x_{ijn} QTY_{ajn} \leq AVAIL_{ai}, \forall a, i \tag{15}$$

$$y_{ij} = \sum_n x_{ijn} \tag{16}$$

$$x_{ijn} \in \{0,1\}, \forall i, j, n \tag{17}$$

This problem can now be decomposed by interaction prediction. If we take *y*, *x*, and λ to be given at the Master level since they are solved by the subproblem, the above simplifies to determining the target-to-base assignments σ_{ij}

maximize$_\sigma$ $$\sum_i \sum_j \sum_n PREF_{jn} x_{ijn} + \lambda_{ij}(\sigma_{ij} - y_{ij}) \tag{18}$$

subject to $$\sum_i \sum_j \sigma_{ij} \leq 1, \forall j \tag{19}$$

$$\sigma_{ij} \in \{0,1\} \tag{20}$$

The solution technique employed to solve this master-level problem must recognize the fact that *y* and λ are functions of σ. That is, σ should be updated so that the solution converges; if the master level problem is solved treating *y* and λ as fixed constants at each iteration, the procedure is likely to cycle.

Given a value for σ, the subproblems solve for *y*, *x*, and λ. The resulting formulation for base subproblem *i* is

minimize$_\lambda$ maximize$_x$ $$\sum_j \sum_n PREF_{jn} x_{ijn} + \lambda_{ij}(\sigma_{ij} - y_{ij}) \tag{21}$$

subject to $$\sum_n \sum_j x_{ijn} QTY_{ajn} \leq AVAIL_{ai}, \forall a \tag{22}$$

$$y_{ij} = \sum_n x_{ijn} \tag{23}$$

$$x_{ijn} \in \{0,1\}, \forall j, n \tag{24}$$

The subproblems thus determine not only how to strike their allocated targets, but through λ, communicate the sensitivity of their solutions to the assignment of additional targets or the removal of existing target assignments.

This information is used in the following iteration at the master level to derive the next σ. This process iterates between the Master and subproblems until a defined stopping criterion is reached (e.g., diminishing returns).

5.1.4.2 Detailed Example: *Interaction Prediction Decomposition of Simplified Formulation*

In this section, we employ a simpler version of the full problem in order to illustrate the mechanics of developing and solving the multilevel, decomposed problem. Rather than try to decompose the full air operations model as defined above, we have chosen to use an abstraction wherein many of the details used by lower-level planning are eliminated while still retaining the key characteristics of the air operations planning problem.

The scenario for this problem includes a set of two air bases, located in the region of interest, from which sorties can be launched. Each air base is assigned a priori a set of resources consisting of strike and escort aircraft from which strike packages can be composed. In addition to air bases, there are targets that are distributed throughout the geographic region and protected by enemy air defense. Targets each have a value determined by the Commander's Intent based on geographic region, functional category, and scenario objectives. All the constraints mentioned above for air operations planning also apply to this scenario. For this simplified problem, we restrict strike packages to be formed only with aircraft from the same air base. Each target can be assigned to only a single subproblem (base). The objective of the scenario is to use the resources available at each of the air bases to determine the best plan to prosecute targets to maximize target value achieved.

There are a large number of targets and only a limited number of aircraft at each base. Together, the two levels of the planner must determine which targets are to be attacked, which air base should be assigned to attack the target, and which aircraft are to be assigned for each attack. With minor exceptions, the problem just described is similar to the description of the knapsack problem: given a fixed-volume knapsack and a set of items of different sizes, each worth a different amount, determine the optimal packing strategy for the knapsack such that the total value of the items placed in the knapsack is maximized. By setting aside the scheduling aspect of the problem, we can formulate our model as a multiple knapsack problem, each base corresponding to one of the knapsacks.

Each air base develops a plan for attacking its assigned targets; however, because of the large number of targets, it is not possible to include all targets in the plan. Thus, there is an analogy between the targets placed into an air base's plan and items being stuffed into a knapsack. Targets, like items, are each worth a different amount, each requiring the expenditure of resources. Each air base, like a knapsack, has a limited set of resources for pursuing targets. The objective is to maximize the total target value over the plans of all the air bases while using only the fixed set of resources given to each base. When the problem is decomposed using the interaction prediction methodology, each of the subproblems takes the form of single knapsack problem. The master problem becomes an allocation problem. It is the master-level planner's responsibility to determine how best to allocate targets to individual subproblems such that the coordination of the resulting plans gradually reaches optimality over the course of several iterations (here, we will refer to those iterations as *negotiation cycles*).

In this formulation, we relax the binary constraints on targets, thereby allowing air bases to schedule fractional flights to prosecute fractional targets. This will allow us to simplify our discussion of the application of interaction prediction decomposition to the problem. We do, however, keep the constraints that require that each target be assigned to only a single air base. We shall see that by doing so, the undecomposed problem becomes a mixed integer program. The decomposed problem will have linear programs as subproblems and a discrete assignment problem at the master level. The full formulation of the problem is given below.

5.1.4.2.1 Simplified Problem Formulation

Let x_i represent a vector of decision variables corresponding to the set of targets to be attacked by base i. If base i is to attack target j, then x_{ij} is 1, otherwise it is 0. The decision vectors for each air base have the same dimension, corresponding to the total number of targets. For example, for a scenario with four targets, if

$$x_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

we interpret this as air base 1 will attack targets 1, 3, and 4, and will not attack target 2.

Let V_i be a vector whose elements represent the overall value of each target to each air base. We assume that the elements of V_i represent for each target the combined measure of the target's value per the Commander's Intent, as well as the risk and cost of prosecuting the target from air base *i* in terms of weapons used, fuel used, and aircraft flight time. By including cost in these coefficients, we choose the lowest cost plan among the ones that achieve the same target value.

Let C_i be a matrix whose elements define the aircraft resources required to form packages to strike each target from base *i*. The j$^{\text{th}}$ column of C_i represents the number of aircraft of each type required to form a package to attack target *j* from base *i*. We assume for simplicity that there is only a single aircraft package per target per air base. For example, for base 1

$$C_1 = \begin{array}{l} \begin{bmatrix} 1 & 2 & 4 \\ 2 & 2 & 2 \\ 4 & 4 & 5 \end{bmatrix} \end{array} \begin{array}{l} \textit{weasel} \\ \textit{jammer} \\ \textit{striker} \end{array}$$

indicates that the package of aircraft required to attack target 1 (column 1 of C_1) consists of one weasel, two jammers, and four strikers. The package of aircraft required to attack target 2 consists of two weasels, two jammers, and four strikers. The package of aircraft required to attack target 3 consists of four weasels, two jammers, and five strikers.

Let K_i be a vector whose elements represent the aircraft resource constraints for air base *i*. It consists of the number of each type of aircraft available at air base *i*.

Given these definitions, we model the simplified air strike operations problem as

$$\max \sum_{\mathrm{i}=1}^{\mathrm{n}} \overset{\varpi}{\mathrm{V}}_{\mathrm{i}}^{T} \overset{\varpi}{x}_i$$

Subject to

$$\sum_{i=1}^{n} \overset{\varpi}{x}_i \leq \overset{\varpi}{1} \qquad \textit{Each target attacked only once}$$

$$C_i \overset{\varpi}{x}_i \leq \overset{\omega}{K}_i, \forall i \qquad \textit{Use only available aircraft}$$

$$\overset{\varpi}{x}_i \geq \overset{\omega}{0} \qquad \textit{Positivity constraint}$$

The first constraint states that the sum across bases of the planned attacks on each target totals 1. This is the constraint that couples the planning for strike packages across bases, and is the one that we focus on in decomposing the problem into decoupled base subproblems.

The second constraint is a substitute for all the side constraints that would appear in a more complete formulation of an air operations planning problem. For our simple problem, these are directly analogous to the packing constraints of a knapsack problem and are referred to here as the resource constraints. The sum of the aircraft resources used to attack targets from each air base must not exceed the total resources, K_i, available at each air base. The cost matrix, C_i, consists of the aircraft types required to form a package for each target. In a more complete formulation, the cost matrix and the vector of total resources would consist of complex functions that incorporate all the details, interrelationships, and logistical costs of forming packages.

The final constraint is a non-negativity constraint stating that the target selection decision variables must be positive.

5.1.4.2.2 Overview of Interaction Prediction Decomposition Methodology

1. Identify variables in either/or the objective function and constraints that cause a coupling of what would otherwise be decoupled optimization problems.
2. Assume that a master level sets and iteratively updates values for those coupling variables. Setting those values results in decoupled (sub)problems that can be solved independently given values of the coupling variables.
3. Write a Lagrangian representation of the full problem as a sum of Lagrangians, each representing a subproblem.
4. Use the subproblem Lagrangians to identify the (sub)optimization problems.
5. Develop approaches to solving the subproblems.
6. Substitute values of the decision variables and Lagrange multipliers (shadow prices) produced as solutions to each of the subproblems as known values into the full Lagrangian, with the result that only the decoupling variables are unknown. The master optimization problem is identified from this Lagrangian and solved to produce improved values for the decoupling variables.
7. Iterate between (5) and (6) (see Figure 5.1-1) until a stopping criterion (e.g., diminishing returns) is satisfied. These iterations are guaranteed to converge to the (undecomposed) optimal solution under specified smoothness and convexity assumptions [1].

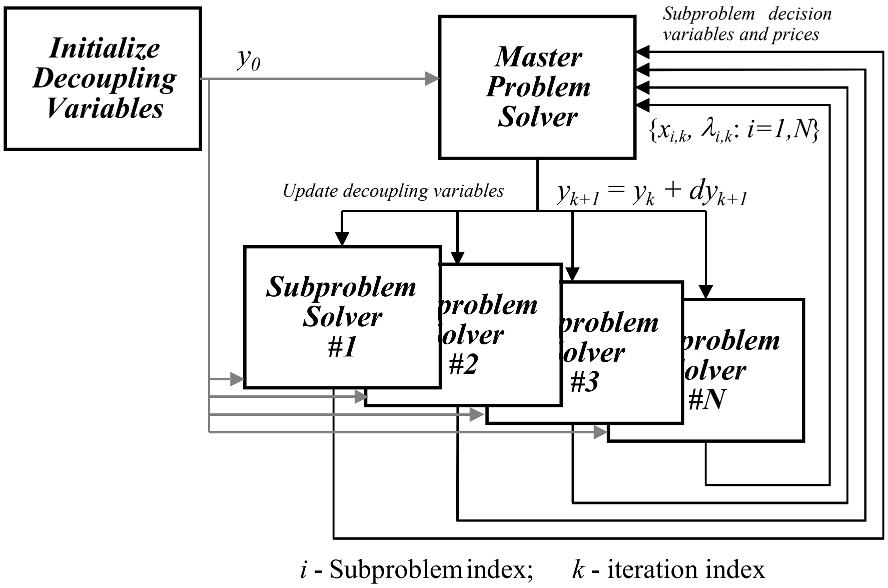

Initialize Decoupling Variables
y_0
Master Problem Solver
Subproblem decision variables and prices
$\{x_{i,k}, \lambda_{i,k}: i=1,N\}$
Update decoupling variables
$y_{k+1} = y_k + dy_{k+1}$
Subproblem Solver #1
problem olver #2
problem olver #3
problem olver #N
i - Subproblem index; k - iteration index

Figure 5.1-1. Iterative solution to decomposed problem.

Negotiation Interpretation

Each subproblem, in effect, tells the master: "For the values of the decoupling variables (y) that you have given me, here is the best solution (x_i) that I can provide along with associated shadow prices (λ_i)." The master uses those decision variables and prices to update the values of the decoupling variables in an attempt to improve the overall solution across all subproblems on the next iteration. We refer to this iterative process as a negotiation in the sense that the subproblems are saying: "Here is the best I could do, and with a change in the coupling variables, I can do this much better." Another interpretation is that the master starts with a poor model of the subproblems' capabilities and guesses at an initial value of the decoupling variables. As the iterations progress, the master's models improve, allowing it to make improved choices for the decoupling variables.

5.1.4.2.3 Simplified Problem Decomposition

In order to decompose the simplified air strike operations problem, we introduce binary decoupling variables, y_i, to represent a discrete assignment of targets to each air base i. These decoupling variables are defined at the master level as indicated in Figure 5.1-1 and passed to what we will develop as lower level, base, subproblems. When an element of y_i is set to 1, that implies that air base i has been assigned the corresponding target. Since we have assumed that the only resources available to each subproblem are those aircraft currently stationed at the corresponding air base (i.e., K_i is fixed), the aircraft available in each subproblem determines the capability and, ultimately, the effectiveness of each subproblem against the targets assigned to it. For example, a target set assigned to an air base with no bomber aircraft would not perform as efficiently against targets that require bombers as opposed to a base that has bomber resources. Therefore, the task of the master-level planner is to allocate targets to subproblems, based on their abilities and available resources, to achieve the best overall plan across all bases. This is accomplished by setting the y_i's correctly.

Full Formulation with Decoupling Variables

The first step in developing the decomposition is to introduce the decoupling variables, y_i, in order to replace the first constraint in the original formulation, $\sum_{i=1}^{n} \overset{\varpi}{x}_i \le \overset{\varpi}{1}$, with the following two constraints

$$\sum_{i=1}^{n} \overset{\varpi}{y}_i = \overset{\varpi}{1}$$

$$\overset{\varpi}{x}_i \le \overset{\varpi}{y}_i, \quad \forall \mathrm{i}$$

The first ensures that each target is only assigned to a single air base. The second set of constraints specifies which targets each air base is allowed to attack. Since a base may not have sufficient aircraft resources to attack all assigned targets, the inequality in the second constraint gives a base the freedom to choose not to attack an assigned target.

Incorporating the decoupling variables and rewriting the problem in standard form gives

$$\min \ -\sum_{\mathrm{i}=1}^{\mathrm{n}} \overset{\varpi}{\mathrm{V}}_{\mathrm{i}}^{T} \overset{\varpi}{x}_i$$

$$C_i \overset{\varpi}{x}_i \le \overset{\omega}{K}_i, \forall i \qquad \textit{Use only available aircraft}$$

$$\overset{\varpi}{x}_i - \overset{\varpi}{y}_i \le \overset{\omega}{0}, \quad \forall \mathrm{i} \qquad \textit{Attack only assigned targets}$$

$$\sum_{i=1}^{n} \vec{y}_i = \vec{1}$$ *Attack each target only once*

$$\vec{x}_i \geq \vec{0}, \vec{y}_i \in Binary$$

Subproblem Formulation

The subproblem formulation is based on the assumption that the y_i's defined by the master are known by and fixed for each subproblem. Given that assumption, the full formulation can be deomposed into n separable subproblems, one for each base i

$$\min \ -\vec{\mathrm{V}}_{\mathrm{i}}^{T} \vec{x}_i$$

Subject to

$$(\vec{x}_i - \vec{y}_i) \leq \vec{0}$$

$$(C_i \vec{x}_i - \vec{K}_i) \leq \vec{0}$$

$$x_i \geq 0$$

Thus, each base subproblem can be solved as a linear program. The value of the complete objective function for any given value of the y_i's is simply the sum of the individual objective functions of the subproblems.

Master Problem Formulation

To develop the master problem formulation, which when solved provides values for the decoupling variables, y_i's, we begin by forming the Lagrangian for the full formulation by dualizing the constraints employed in the subproblem

$$L = \sum_{i=1}^{n} \left[-\vec{V}_i^T \vec{x}_i + \lambda_i^T (\vec{x}_i - \vec{y}_i) + \xi_i^T (C_i \vec{x}_i - \vec{K}_i) \right]$$

Recall that we assume that at the master level, the solution to the subproblems is known. Note that these solutions include, in addition to the target assignments, the dual variables associated with the subproblem constraints, λ_i and ξ_i. In order to determine new values for the decoupling variables, we form the total differential of the Lagrangian with respect to those variables

$$dL = \sum_{i=1}^{n} -\lambda_i^T d\vec{y}_i$$

This establishes a relationship between the changes in the decoupling target assignments and the changes in the Lagrangian. The objective of the master problem is to choose changes, dy_i, in the decoupling variables that reduce the value of the Lagrangian (since we are attempting to minimize) while honoring the constraint that the updated values of the coupling variables satisfy the constraint that all targets are assigned and that no target is assigned to more than one base: $\sum_{i=1}^{n} \vec{y}_i = \vec{1}$.

With the dual variables, λ_i, provided by the subproblems, the master can update the decoupling variables as follows. First note that since the sum of each of the elements of the y_i's must equal 1, and since they are binary, the only values for the elements of the dy_i are –1, 0, or +1 (i.e., remove the target from the subproblem, leave the target in the subproblem, add the target to the subproblem). The optimal strategy for reducing L (i.e., choosing the elements of the dy_i to maximize the reduction in dL) is to choose dy_i that result in reassigning targets from their current subproblems to subproblems with the larger associated elements of λ_i. A numerical example is provided later to make this more concrete.

Marginal Utility Interpretation

Given values for the y_i, the Lagrangian L can be separated into elements associated with each of the subproblems L_i

$$L_i(\overset{\varpi}{x}_i,\overset{\omega}{\lambda}_i,\overset{\omega}{\xi}_i;\overset{\varpi}{y}_i) = \text{-}\overset{\omega}{\mathrm{V}}_{\mathrm{i}}^{T}\overset{\varpi}{x}_i + \lambda_i^T(\overset{\varpi}{x}_i - \overset{\varpi}{y}_i) + \xi_i^T(C_i\overset{\varpi}{x}_i - \overset{\omega}{K}_i)$$

One of the Karush-Kuhn-Tucker (KKT) conditions provides a way to update the decoupling variables. For each subproblem, the condition holds for nonzero elements of x_i

$$\frac{\delta L_i}{\delta \overset{\varpi}{x}_i} = -\overset{\varpi}{V}_i + \overset{\varpi}{\lambda}_i + C_i^T\overset{\varpi}{\xi}_i = \overset{\varpi}{0}$$

The condition states that for nonzero elements of the decision variables x_i there is a corresponding set of relationships among the dual variables, namely

$$\overset{\omega}{\lambda}_i = \overset{\omega}{V}_i - C_i^T\overset{\omega}{\xi}_i$$

In the following, we use this expression to interpret the master problem's strategy for reducing L described above. Given the expression above for λ_i, the master problem can adopt a strategy that is equivalent to swapping targets from one subproblem to the one with the highest associated element of (V_i-$C_i\xi_i$). The elements of V_i-$C_i\xi_i$ can be interpreted as the marginal utility gained/lost for receiving/losing a target. The marginal utility of a target is the true value of the target, $\mathrm{V_i}$, reduced by the opportunity value, $\mathrm{C_i\xi_i}$, the value that could have been gained by applying those resources toward other targets assigned to the subproblem. Essentially, the master planner decides whether to reallocate targets based on the comparisons of how marginally valuable each of the targets are to each of the subproblems. If the master planner sees that another subproblem is able to extract more value from a target than the subproblem that the target is currently in, it will make a swap.

Numerical Example

For this numerical example, we have two air bases, each with identical sets of aircraft resources. There are 10 targets in the scenario, and each air base has a distinct aircraft package that is capable of attacking each target.

The problem is as follows:

The objective function: $\min\ \text{-}\overset{\omega}{\mathrm{V}}_{\mathrm{i}}^{T}\overset{\varpi}{x}_i$

$$\max[11 \quad 5 \quad 10 \quad 9 \quad 10 \quad 0 \quad 20 \quad 0 \quad 1 \quad 7]\overset{\varpi}{x}_1 + [22 \quad 1 \quad 5 \quad 23 \quad 9 \quad 10 \quad 1 \quad 10 \quad 3 \quad 10]\overset{\varpi}{x}_2$$

where the coefficients for each vector, V_i, represent the expected net value (target value minus expected cost of prosecuting the target) accrued for successfully attacking each target from base i.

The resource constraints: $C_i\overset{\varpi}{x}_i \le \overset{\omega}{K}_i, \forall i$

$$\begin{bmatrix} 2 & 2 & 2 & 3 & 4 & 2 & 2 & 6 & 2 & 4 \\ 2 & 2 & 2 & 2 & 2 & 3 & 0 & 2 & 2 & 2 \\ 1 & 2 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 0 \end{bmatrix}\overset{\varpi}{x}_1 \le \begin{bmatrix} 15 \\ 7 \\ 5 \end{bmatrix}\begin{array}{l}\text{Strikers}\\ \text{Jammers}\\ \text{Weasels}\end{array}$$

$$\begin{bmatrix} 2 & 4 & 4 & 2 & 4 & 1 & 5 & 3 & 3 & 2 \\ 2 & 0 & 0 & 2 & 2 & 2 & 3 & 2 & 5 & 1 \\ 0 & 1 & 2 & 2 & 3 & 3 & 1 & 2 & 4 & 1 \end{bmatrix}\overset{\varpi}{x}_2 \le \begin{bmatrix} 15 \\ 7 \\ 5 \end{bmatrix}\begin{array}{l}\text{Strikers}\\ \text{Jammers}\\ \text{Weasels}\end{array}$$

Recall that the columns of C_i represent the package composition for each of the targets to be attacked from base *i*.

The initial allocation of targets (the master problem's initial guess) can be arbitrary as long as each target is assigned to a single air base. We start with the following initial target assignment

$$\overset{\varpi}{y}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \text{ and } \overset{\varpi}{y}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

The assignments are passed to the subproblems for detailed planning. The subproblems return an initial set of decision variables (targets to be attacked) representing a combined total objective value of 57. That initial solution is

$$\overset{\varpi}{x}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ and } \overset{\varpi}{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The dual prices returned for each of the resource constraints per subproblem are

$$\overset{\varpi}{\xi}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \overset{\varpi}{\xi}_i = \begin{bmatrix} 0 \\ 0 \\ 2.5 \end{bmatrix}$$

The master level uses these values to evaluate the marginal utility values, *V-Cξ,* to determine how to reallocate targets to improve the overall plan. We list the *V-Cξ* results below for each target in each subproblem. Here *Owner* refers to the base to which each target is currently assigned

	Subproblem1	Subproblem2	Owner
Target1	11	22	2
Target2	5	0	1
Target3	10	0	2
Target4	9	18	1
Target5	10	1.5	2
Target6	0	2.5	1
Target7	20	0	2
Target8	0	5	1
Target9	1	0	2
Target10	7	7.5	1

Based on these marginal utility values, the master swaps two targets. Target 3 is currently assigned to subproblem 2 and has a marginal utility of 0. Additional value would be gained if it were reassigned to subproblem 1, which values target 3 at 10. The same is true for target 7, which is greatly valued by subproblem 1. The master planner reallocates these targets to the appropriate subproblem by updating the decoupling variables

$$\overset{\varpi}{y}_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \text{ and } \overset{\varpi}{y}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

This allocation results in an improvement of the objective value from 57 to 87. Iterating these steps results in a gradual improvement of the solution toward optimality.

Iteration #	Objective Value
0	57
1	81
2	96
3	104.5
4	110
5	110*

Negotiation is halted after five iterations, since the comparison of marginal utility values indicates that it is not possible to improve the solution further. As a basis for comparison, we have solved the full mixed-integer programming (MIP) formulation, where target planning and target allocation were solved undecomposed. The optimal MIP solution and optimal negotiation solutions are identical. Table 5.1-1 shows the solution vectors for each subproblem per iteration, with the last column being the solution to the MIP problem.

Table 5.1-1. Solution Vector for Each Subproblem per Iteration.

Iteration 0		Iteration 1		Iteration 2		Iteration 3		Iteration 4		Iteration 5*		MIP Optimal	
SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2	SP1	SP2
0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0.5	0	1	0	0.5	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	0	1	0	1	0	1	0	1	0	1
0	1	0	1	0	0.333	1	0	1	0	1	0	1	0
0	0	0	0	0.333	0	0	0.333	0	0	0	0	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	0	0	0	1	0	1	0	1	0	1	0	1
0	0	0	0.5	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	0	1	0	1	0	1

5.1.4.3 Price-Coordinated Decomposition of Math Programming Formulation

For price-coordinated decomposition, we do not need the σ dummy variables. Instead, we start with the original formulation and dualize the coordinating constraint. The result is

$$\min_{\mu} \max_{x} \quad \sum_{i}\sum_{j}\sum_{n} PREF_{jn} x_{ijn} + \sum_{j} \mu_j \left(1 - \left(\sum_{i}\sum_{n} x_{ijn} \right) \right) \tag{25}$$

$$\text{subject to} \quad \Sigma_{n} \Sigma_{j} x_{ijn} QTY_{ajn} \leq AVAIL_{ai}, \forall a \tag{26}$$

$$x_{ijn} \in \{0,1\}, \forall j, n \tag{27}$$

$$\mu_j \geq 0, \forall j \tag{28}$$

In the price-coordinated decomposition, the μj represent the value that some subproblem can achieve by hitting target j. The master level according to the minimization sets these prices

$$\text{minimize}_{\mu} \sum_{j} \mu_j \left(1 - \left(\sum_{i}\sum_{n} x_{ijn} \right) \right) \tag{29}$$

$$\mu_j \geq 0, \forall j \tag{30}$$

The solution method employed at the master level must recognize that x is a function of μ. The master level problem is updated at each iteration using the results of the subproblems in the previous iterations.

Given the μ, each base subproblem i solves the following maximization problem

$$\text{maximize}_{xi} \quad \Sigma_{j} \Sigma_{n} PREF_{jn} x_{ijn} + \mu_j \left(1 - \left(\Sigma_{n} x_{ijn} \right) \right) \tag{31}$$

$$\text{subject to} \quad \Sigma_{n} \Sigma_{j} x_{ijn} QTY_{ajn} \leq AVAIL_{ai}, \forall a \tag{32}$$

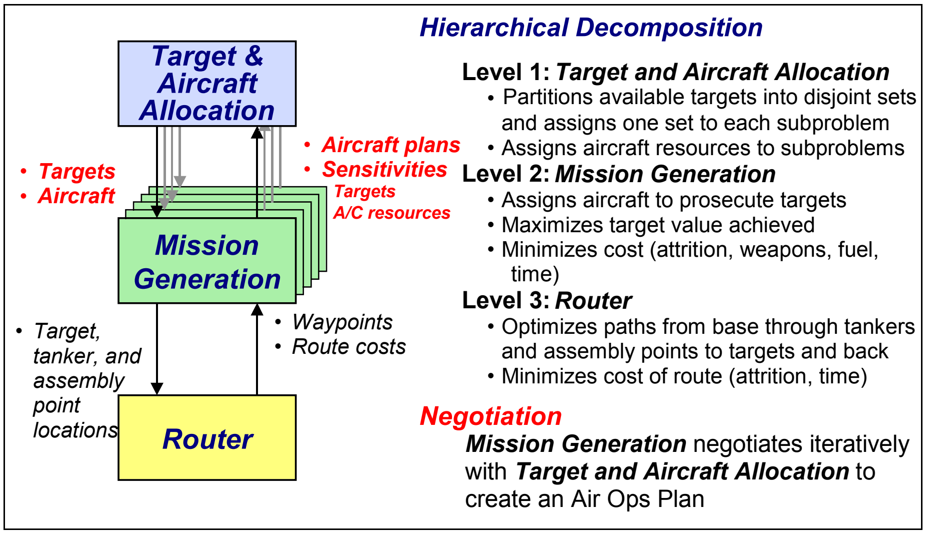
$$x_{ijn} \in \{0,1\}, \forall j, n \tag{33}$$

Thus, subproblem i must effectively justify its choice to hit target j by paying (through the penalty term in the objective function) the dual cost. If the dual cost is too great, that signifies that another base is better able to hit the target, so base i does not include it in its solution. If, on the other hand, base i is able to hit the target better than other bases, it will at the optimal solution include target j and generate more value in so doing than it must pay through the dual cost.

5.1.5 Algorithms

The levels of the decomposed strike-planning problem are depicted in Figure 5.1-2. The *Target and Aircraft Allocation* level assigns (decoupling variables) the targets to be pursued and the aircraft resources to be used by the *Mission Generation* planners. The *Mission Generation* planners assign and schedule specific aircraft package and weapon resources for specific targets. The *Mission Generation* decision-making is supported by the *Router*, which, given (decoupling variables) specific target locations, assembly points, tanker locations, and aircraft package composition provides attrition, time and fuel, costs, accounting for jammer and escorts per the package definition and constrained by assembly points and tanker locations, as specified by the *Mission Generation* planner, as well as by available fuel and aircraft performance.

Both optimal and heuristic solutions have been developed to solve these optimization problems as described in the following sections. The heuristic planners were found to be useful for both early development of the controller architecture and as a basis of comparison with the performance of the optimal solution, with the expectation that the objective values produced by heuristic planners will follow trends similar to those of the optimal controller. Indeed, the heuristic solutions were sufficiently effective that elements of them have become part of the final controller. In



Hierarchical Decomposition

Level 1: ***Target and Aircraft Allocation***

- Partitions available targets into disjoint sets and assigns one set to each subproblem
- Assigns aircraft resources to subproblems

Level 2: ***Mission Generation***

- Assigns aircraft to prosecute targets
- Maximizes target value achieved
- Minimizes cost (attrition, weapons, fuel, time)

Level 3: ***Router***

- Optimizes paths from base through tankers and assembly points to targets and back
- Minimizes cost of route (attrition, time)

Negotiation

Mission Generation negotiates iteratively with ***Target and Aircraft Allocation*** to create an Air Ops Plan

Figure 5.1-2. Three-Level Decomposition.

particular, solving the decomposed problem requires several iterations (negotiation exchanges) between the Level-1 and Level-2 planners in order to approach optimality. The initial iterations are performed with heuristic Level-1 and Level-2 solutions, which are much faster than the optimal. In the final iterations, the optimal solutions can be used to replace the heuristics. This can be viewed as providing a "warm start" for the optimal solutions. Furthermore, as we will see, the heuristic Mission Generation planner is used to develop "composite variables" for the optimization-based solution.

5.1.5.1 Heuristic Algorithms for Levels 1 - 3

The ***heuristic target partitioning (Level-1) planner*** allocates targets to the Mission Generation (Level 2) subproblems to encourage the generation of plans that prosecute high-valued targets in a timely manner while maintaining workload balance among subproblems. A target list is created and ranked in order of decreasing target value. The Level-1 planner allocates each target on the target list to the nearest base, starting with the highest-valued target, thereby allowing higher-valued targets to be prosecuted sooner. Workload balance is maintained by ensuring that the number of targets allocated to each base is proportional to that base's workload capacity. The workload capacity is measured as a function of that base's weapon delivery capacity.

In addition, the Level-1 planner is capable of negotiating with the lower-level planner to improve on the target allocation. Negotiation is an iterative process in which the lower levels pass sensitivity information and completed plans to the master level. The master level processes the sensitivity information to determine a new allocation of targets and resources to subproblems. These modified subproblems are returned to the lower levels for planning, resulting in plan improvements.

The ***heuristic Mission Generation planner*** maximizes target value specified by region and functional category as established via the Commander's Intent Input Matrix while simultaneously attempting to minimize the total cost comprising:

- Attrition risk.
- Cost of time.
- Weapon utilization.
- Mission retasking cost.

The heuristic implementation of the Level-2 planner provides a baseline controller for comparison with experiments of the optimal integer programming solution. In addition, as described above, it may be used in conjunction with the optimal planner to speed convergence by providing a warm start. It sequentially constructs strike packages, assigning aircraft to targets and optimizing the incremental contribution to an objective function for each aircraft mission that the planner generates. This incremental approach has been found to generate reasonable plans very quickly. The optimization accounts for:

- Target prioritization.
- Assignment of aircraft to targets.
- Weaponeering via prespecified package configurations.
- Asynchronous scheduling of sorties with package-level synchronization of time on target.

The Level-3 planner is a ***Strategic Router*** that supports the Level 2 planner by providing the cost of constrained minimum risk routes for specified aircraft-target pairs.

The route-planning problem is:

Given a set of

1. **Mission parameters** including:
    - Start location (base or en route) and return base.
    - Required ingress and/or egress assembly points.
    - Target location.
    - Set of all tanker locations.
2. **Aircraft parameters** including:
    - Fuel endurance.
    - Pilot endurance.
3. A **Threat model** including:
    - Threat density.
    - Detection range.
    - Likelihood of engaging.
    - Possibility of attrition in an engagement (P_k)
4. **Escort level** representing one of:
    - No escort.
    - Wild weasels.
    - Weasels plus escort jammers.

Determine a strategic-level route (10- to 30-km grid spacing for waypoints) that:

- Is feasible with respect to aircraft fuel endurance.
- Minimizes risk from threat engagement.
- Allows for two refueling activities on each segment.
- Adheres to specified assembly points.

An A* search-based router has been implemented and integrated with the Level-2 planner. Figure 5.1-3 illustrates the results from the router for a 30-km gridding with a required assembly point on ingress. The legend indicates

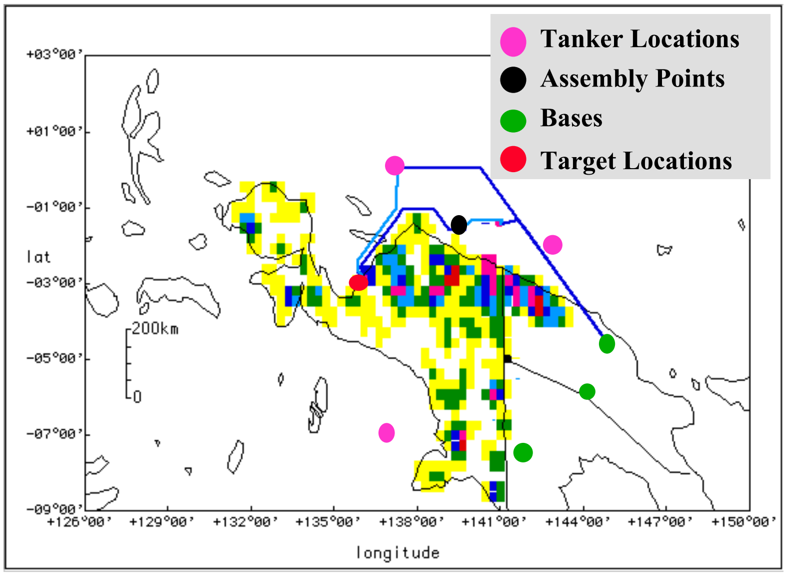


Figure 5.1-3. Example Routing with Required Assembly Point on Ingress.

base, tanker, target, and assembly point locations. Grid threat levels are color coded from low (yellow) to red (high).

5.1.5.1.1 Master-Level Heuristic Planner

The Level-1, or master-level planner, is invoked once per planning cycle and is responsible for generating subproblems for the Mission Generation (Level-2) planner to solve. Each subproblem consists of a set of targets, a set of aircraft, an objective function, and a set of constraints. The targets and aircraft allocated to each subproblem inherently determine the objective function and constraints for the subproblem.

Targets each have a property known as target value, defined via the Commander's Intent value model. The target value expresses a target's potential contribution toward achieving the overall campaign objectives if it is destroyed. A subproblem's objective function is the sum of the expected target value to be achieved in the planning cycle. In other words, a subproblem's objective function can claim a target's value only if a plan to attack the target is generated for the current planning cycle. The constraints of a subproblem are defined by the scenario definition (e.g., the threat regions place constraints on escort requirements for strike packages) and by the interaction between targets and aircraft in the subproblem (e.g., weaponeering constraints specify the number and types of weapons an aircraft needs to achieve a certain level of target damage against a specific target type). In general, the resources allocated to a subproblem place constraints on the lower level planner, which affect its planning and selection of targets.

Ideally, the master-level planner should first analyze the scenario and determine the number of subproblems required. It then needs to allocate resources and targets to each of these subproblems to facilitate their solution in a timely and effective manner by the Mission Generation planner.

Figure 5.1-4 shows the layout of the standard scenario used for testing. There are five air bases, each indicated by an arrow. The targets are concentrated in the western half of the island and are circled. The layout of the test scenario suggests a natural decomposition of the problem geographically by air base. The heuristic algorithm used in the master-level planner to generate in initial target allocation produces such a geographic decomposition. When the master planner is invoked, it creates a separate subproblem for each air base in the scenario. The resources

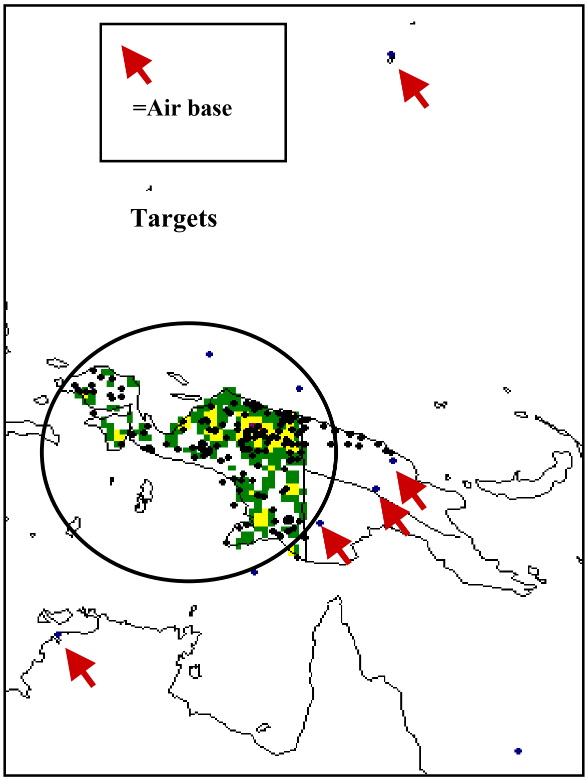


Figure 5.1-4. Layout of Standard Scenario for Testing.

(aircraft) applied to each subproblem are determined automatically. Since aircraft each have a default air base assignment, initially determined by the scenario definition, the master-level planner nominally assigns each aircraft to the subproblem corresponding to its default air base.

With the resource assignments determined by this natural heuristic decomposition, the main task of the master-level planner becomes one of target partitioning. The master-level planner uses its high level understanding of the capabilities of the resources in each subproblem and the overall objective of the campaign to determine a distribution of targets to balance the subproblem workloads and to maximize the rate at which the goals of the campaign are reached. Each target is considered individually and is assigned to the subproblem that will make the best use of it. Note that a target may only be assigned to a single subproblem. This is to prevent squandering resources when multiple subproblems pursue the same target within the same planning cycle. Figure 5.1-5 shows one possible outcome of the master-level target partitioning algorithm. Each of the rectangles represents a set of targets. An arrow emanating from each rectangle indicates the subproblem (air base) to which the target set is allocated.

Workload Balance

As stated previously, the objectives of the master-level planner are to balance the workload of each subproblem and to maximize the number of high-valued targets prosecuted over the course of the campaign. The planner seeks to minimize the inactivity of each subproblem at any point in time by distributing the workload such that the number and type of targets assigned to each subproblem is proportional and suited to the subproblem's target prosecution ability. Subproblems that can handle more targets are given more targets, and vice versa. If the workload were unbalanced, one subproblem might work very hard to prosecute all of its assigned targets, while other subproblems would be idle.

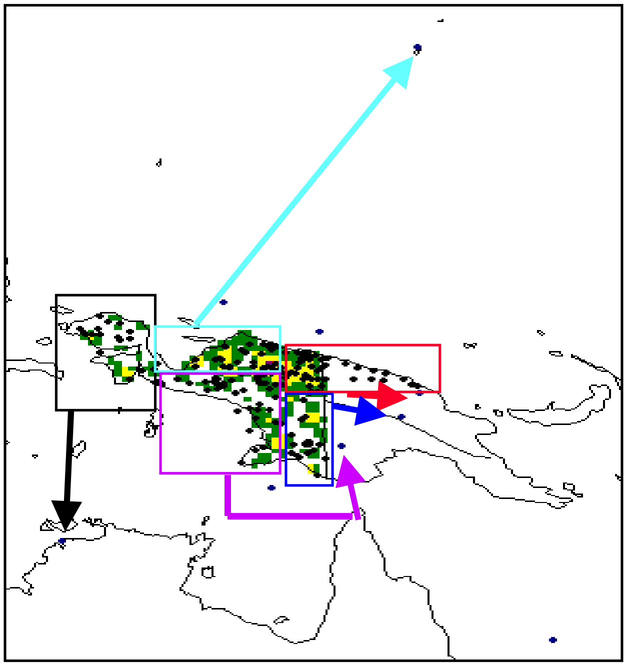


Figure 5.1-5. Possible Outcome of the Master-Level Target Partitioning Algorithm.

At a high level of detail, the master-level planner ignores temporal issues and associates the workload capacity of a subproblem to the number of weapons it is capable of delivering per wave. That is, the workload of a subproblem is equated with the total number of weapons that could be delivered to a target if all the subproblem's aircraft, when fully loaded, were to empty their weapons stores onto the target. In the scenario, each aircraft can be configured with several types of weapons, where the number of weapons depends on the weapon type. The planner's estimate of the subproblem’s weapon-carrying capacity is produced by averaging the number of weapons over all of the possible weapon configurations for each aircraft in the subproblem and totaling the averages over aircraft assigned to the subproblem. This total represents the number of bombs that the subproblem can drop per wave. Notice that this is an optimistic estimate of weapon delivery capability, since discrete strike packages, not individual aircraft, are sent to prosecute targets. It is possible to send a strike package in which only half the aircraft drop weapons. One drawback of the current approach is that it ignores package constraints and inherently assumes that the lower-level planner can form arbitrary strike packages. The following example illustrates this point.

Example

Consider a subproblem defined as two strikers (one striker with two bombs, and another striker with three bombs) and two targets (one target requires four bombs, and the other target requires one bomb). The total number of bombs available per wave is five, and the total number of bombs required to hit the targets is five. From this information, the planner assumes that it can hit both targets, when in fact, the mission generator can only plan for a single target; the lower level planner will either generate a mission to hit the first target with both strikers, or hit the other target with a single striker.

Campaign Rate

The master planner also attempts to maximize the rate at which the campaign goals are achieved (i.e., the rate of target value acquired). To do so, the planner ensures that as many as possible of the high-valued targets are processed first. In the worst case, if all the high-valued targets were assigned to a single subproblem, it would likely not have enough resources to pursue them all. Many targets would not be prosecuted within the planning cycle. It would be more efficient to spread the high-valued targets among all the subproblems so that more of these targets were processed simultaneously, thereby increasing the rate at which target value is acquired.

This distribution of targets is accomplished through presorting all the targets in the scenario according to value from highest to lowest, and assigning each target to the subproblem whose corresponding base is geographically closest to it. By assigning the highest-valued targets first, rather than choosing targets at random or by other means, the planner ensures that the high-valued targets are well spread out across subproblems. The planner also assigns each target to the closest base, which will minimize the time needed for aircraft to reach the target, thus maximizing the rate at which high-valued targets are destroyed. Before the master-level planner assigns a target to a subproblem, it checks to see whether the subproblem has enough of the required weapon type available to prosecute the target. If it does not, the planner will attempt to assign the target to the next closest base. If a target is assigned to a subproblem, the running total of the weapon delivery capability for the subproblem is decreased by the total number of weapons required by the target. Since target allocation is an iterative process, the weapon delivery capability for all subproblems are reset to their original values if targets remain and none of the subproblems have enough weapons remaining to claim additional targets. Once the target allocation is completed, the subproblems are sent to the lower level planner for detailed planning.

One additional task performed by the master-level planning algorithm is keeping track of the subproblem each aircraft was in previously. Once the lower-level planner returns with detailed plans, the master level notes the originating subproblem for each aircraft before sending the plan out for execution. During the next planning cycle, the master-level planner scans through a list and places aircraft currently en route to a target into its previous subproblem rather than to the subproblem corresponding to its default air base (in general, the previous air base will be the default air base unless some form of negotiation is being executed). It also finds the corresponding targets and replaces them into the subproblem they were in previously. This is intended to eliminate forced target reassignment. Since the target partitioning algorithm will give a different allocation each time the state is changed, this mechanism is necessary to maintain consistency. Without this, targets in one subproblem may be allocated to another subproblem in the next planning cycle. This is especially detrimental to aircraft currently en route to a target only to discover it no longer in the same subproblem. These aircraft would either need to be reassigned to a new target, which would incur a replanning cost, or be sent home. In either case, removing targets from a subproblem without a good reason is extremely disruptive and is currently prevented.

Heuristic Target Partitioning Algorithm

- **Initialization**
  - Determine the number of functional air bases in the scenario for the current planning cycle.
  - For each functional air base in the scenario, create a single subproblem.
  - Sort through the aircraft list, and nominally assign aircraft to the subproblem corresponding to the aircraft's default air base. An exception occurs if an aircraft appears on the prior subproblem assignment list. If the aircraft is on the prior assignment list and currently en route to a target, assign it to the subproblem it was in during the previous planning cycle. Create a previous subproblem assignment list for targets to pair the aircraft's subproblem assignment with the aircraft's intended target.
  - In the initial cycle, the prior subproblem assignment list is empty, so each aircraft is assigned to the subproblem corresponding to its default air base.
  - Once the aircraft assignment is complete, average, over the different mission configurations, the number of weapons each aircraft can deliver regardless of weapon type. The weapon delivery capability of each subproblem is the sum over the average capacity of each aircraft in the subproblem currently on the ground.
  - Copy the entire scenario target list to a temporary target list.
  - Sort the temporary target list by target value.

  - Initialize the weapon delivery capability of each subproblem to the values calculated above.
- **Target Partitioning**
  - While the temporary target list is not empty, continue.
    - Iterate through the temporary target list.
      - Check to see if the target is on the previous assignment list. If a match is found, assign the target to the subproblem indicated on the list. Decrement the weapon delivery capability of the subproblem by the number of weapons required by the target. Go on to the next target.
      - If the target was not on the previous assignment list, check whether all of the subproblems have depleted their weapons. If so, reinitialize the weapon delivery capability of all subproblems to their proper values.
        - Assign each target to the subproblem (air base) that is physically closest to the target and meets the weaponeering requirements (i.e., has enough of the proper weapon type). If the closest subproblem does not have the proper weapon type or if it does not have enough weapons, check the next closest subproblem for eligibility until a valid assignment can be made. If an assignment is made, remove the target from the temporary target list.
        - If the target cannot be assigned to any subproblem, move the target to the reserve target list.
        - If a subproblem is assigned a new target, decrement its weapon delivery capability by the appropriate amount.
    - If the reserve target list is not empty:
      - Copy the reserve target list onto the temporary target list.
    - If the reserve target list is empty:
      - Stop, all the targets have been assigned a subproblem.
- **Subproblem Generation**
  - Send the new subproblems to the lower levels.
  - For each aircraft scheduled to attack targets, record its current subproblem assignment in the prior subproblem assignment list.
  - Send plan out for execution

Extensions to the basic master-level planning algorithm have been considered but not implemented. One consideration that has not been addressed in the current algorithm is the temporal aspect of target allocation. Currently, target allocation is based purely on weapon delivery capability with no consideration for flight times to and from the targets. It is possible that a scenario might arise in the current implementation where a single air base, located far from the targets, with a large weapon delivery capability and very few aircraft to deliver them, could be assigned most of the targets. According to the weapon delivery capability, the subproblem is capable of prosecuting many targets. Given a few targets requiring many bombs, this is truly the case. However, this is contrary to the goal of the master-level planner to maximize the rate at which campaign goals are met. This situation can be remedied if the master-level planner takes into consideration the flight and turn times required of each subproblem when deciding on target assignments. Incorporating temporal considerations into the algorithm will give a better estimate of the workload capability of a subproblem and should result in planning improvements.

5.1.5.1.2 Master-Level Negotiation

Negotiation algorithms for the master-level planner have been implemented. As discussed previously, single-pass decomposition often results in the generation of suboptimal subproblems. This is understandable, since the initial allocation of targets and resources is based solely on the master level's high-level representation of the properties of each subproblem. The negotiation schemes implement an iterative process between the master-level planner and the

Mission Generation planner, or sublevel planner, to adjust the assignment of targets and resources among the subproblems to improve the overall objective function. Along with completed plans for each subproblem, the sublevel planner passes additional sensitivity information to the master-level planner. The master planner is then able to process this sensitivity information to make better decisions as to the allocation of targets and resources. Figure 5.1-6 depicts the iterative negotiation process. Two main algorithms were developed for negotiation: target negotiation and resource negotiation. Various negotiation configurations that combined the two algorithms were also implemented.

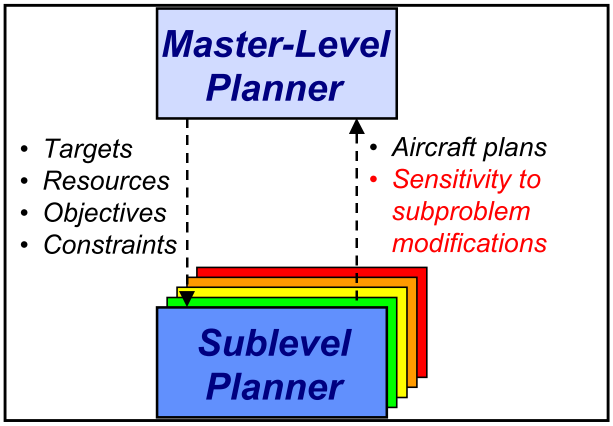
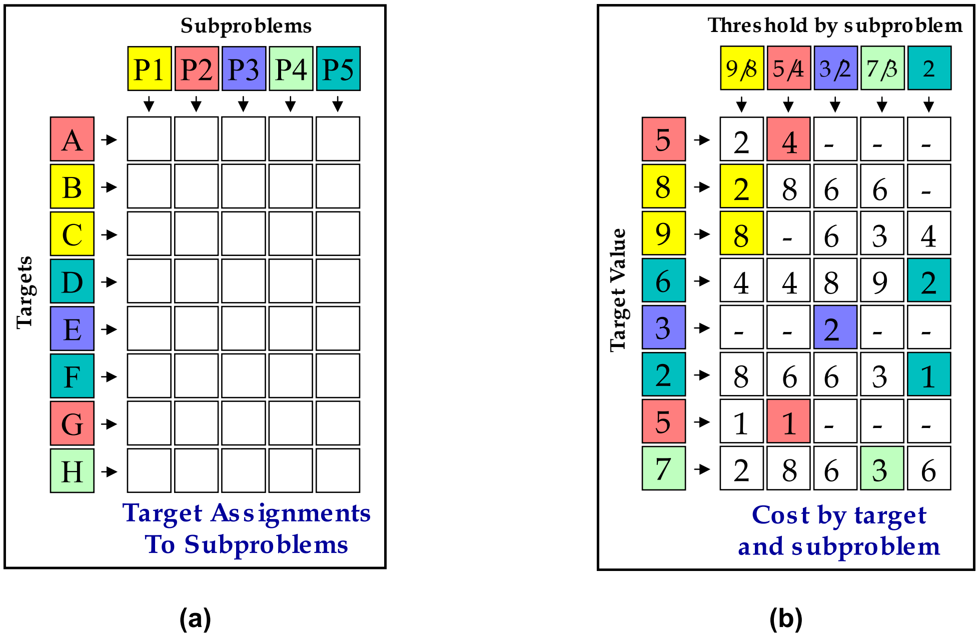


Figure 5.1-6. Iterative Negotiation Process.

5.1.5.1.2.1 Target Negotiation

The master planner uses sensitivity information from each subproblem to reassign the targets among the subproblems. Sensitivity information is a gradient used to express each subproblem's relative affinity for each target in the scenario and the potential objective value improvement it can achieve if allocated a target. The master-level planner calculates the marginal benefit of swapping a target from one subproblem to another (i.e., it determines if the gain in objective value of adding a target into another subproblem more than offsets the loss resulting from removing the target from its current problem).

Target negotiation starts with the master level decomposing the problem in its usual way. Figure 5.1-7(a) shows five subproblems (P1-P5) and eight targets (A-H). The color coding indicates the assignment of targets to subproblems (e.g., targets B and C are assigned to P1). The sublevel planner generates plans for each of the subproblems, and the resulting plans are returned along with sensitivity information. Figure 5.1-7(b) shows a representative example of the data return by the sublevel planner during negotiation. The top row is filled with threshold values, which are estimates of the resource dual values used in interaction prediction. These threshold values are heuristically estimated to be the ratio of value to cost of the last target to be selected by the subproblem, or the lowest ratio of value to cost of any target in each subproblem's plan. The column on the left side indicates the value of each target. The cost for each subproblem to attack each target is given in the data in the matrix. The dashes indicate that the target is infeasible for the subproblem. Note that since each subproblem is only allocated a subset of the targets, the sublevel planner can only get the cost estimates of the targets for the subproblem that it is currently working on (e.g., when SP1 originally returns with a plan, it will only have cost estimates for targets B and C). The cost for targets assigned to the subproblem (e.g., target A is not assigned to SP1) must be estimated in another way. This is done by generating hypothetical subproblems where targets without cost estimates are passed into each of the subproblems one at a time. The reason this in done is to accommodate the way the sublevel planner functions. In order for the sublevel planner to return a cost for a target, a plan must be generated for the target. The sublevel planner is asked to solve the single target subproblem and return a cost estimate to the master level. Notice that since each of these hypothetical subproblems is set up to have all its resources available to prosecute a single



Target Value	9/8	5/4	3/2	7/3	2
5	2	4	-	-	-
8	2	8	6	6	-
9	8	-	6	3	4
6	4	4	8	9	2
3	-	-	2	-	-
2	8	6	6	3	1
5	1	1	-	-	-
7	2	8	6	3	6

Figure 5.1-7. Master-Level Decomposition.

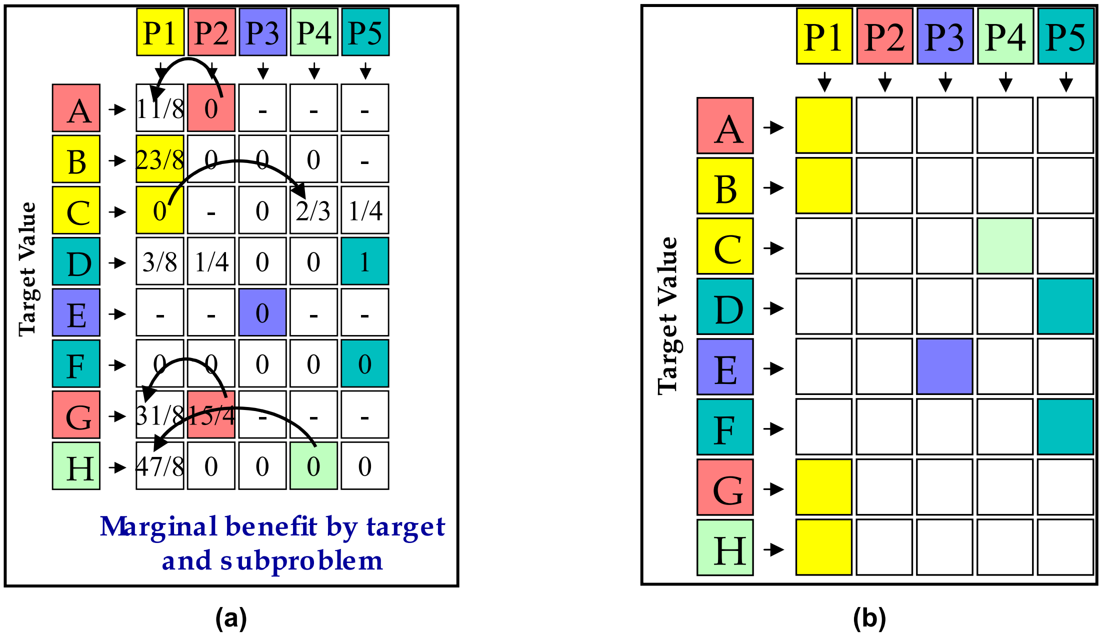
target, the cost estimate is generally optimistic; the sublevel planner always assigns the best resources to minimize the cost. This will not be the case in a real subproblem, where targets must contend with other targets for resources.

The targets selected to be in the plan for each subproblem are color coded to match the corresponding subproblem. Now it is possible to see that the threshold value is indeed the minimum ratio of value to cost of the targets in the plan. For example, the threshold value for subproblem P1 is 9/8 since target C has the lowest ratio of value to cost of the targets (B and C) in the subproblem's plan. The negotiation algorithm processes the sensitivity information to determine the best swaps for improving the global objective value. The marginal value is used to evaluate the importance of each target to its subproblem. The master-level negotiation algorithm calculates the marginal value according to

$$\lambda = \frac{\mathrm{V}}{\mathrm{C}} - T$$

where λ is the marginal value of a target to the subproblem, V is the target value, C is the cost to attack the target, and T is the threshold value. Figure 5.1-8(a). shows the results after performing this calculation for all targets in all subproblems. From the arrows in the diagram, it is possible to see four potential swaps, which will result in a gain in the objective value. The swaps take place between the current subproblem of the target and the subproblem with the highest marginal value for the target. To illustrate, it is beneficial to move target H from subproblem P4 into subproblem P1.

The number of targets to swap per negotiation cycle is known as the step size. The step size should generally be a small percentage of the total number of targets in the scenario (e.g. < 5%) in order to maintain the stability of the algorithm. Larger step sizes may cause cycling, where targets are swapped back and forth between the same subproblems during subsequent negotiation cycles. Figure 5.1-8(b) shows the results of performing all potential target swaps. Once targets have been reallocated, these modified subproblems are sent to the sublevel planners for detailed planning and the process repeats.



	P1	P2	P3	P4	P5
A	11/8	0	-	-	-
B	23/8	0	0	0	-
C	0	-	0	2/3	1/4
D	3/8	1/4	0	0	1
E	-	-	0	-	-
F	0	0	0	0	0
G	31/8	15/4	-	-	-
H	47/8	0	0	0	0

Figure 5.1-8. Reallocation of Targets During Negotiation: (a) Marginal Values Indicate Potential Target Swaps, (b) New Target Allocation after Swapping Targets.

Estimates of many of the parameters used throughout the negotiation process introduce a variety of errors. As such, the global objective value does improve, but is not guaranteed to increase monotonically after each negotiation cycle. Therefore, the master-level planner must keep track of the best plan over all the negotiation cycles, since that is the one that should be executed. The fact that the objective value does not converge precludes using a diminishing return termination function. Instead, the number of iterations is fixed by the user.

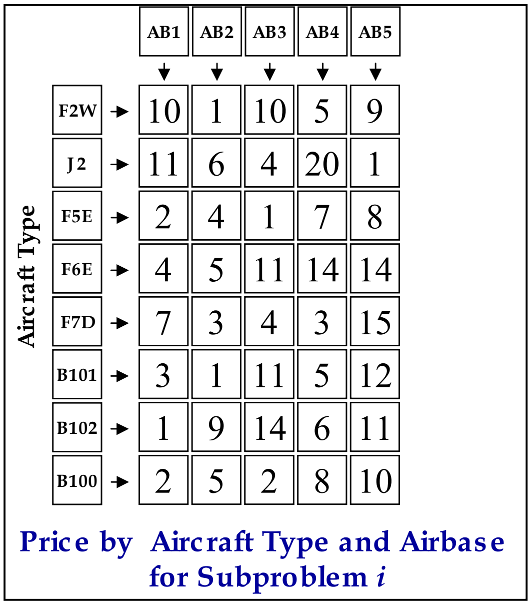
Target Negotiation Algorithm

- Execute the heuristic decomposition algorithm.
- While the maximum number of negotiation iterations has not been reached, continue.
  - After plans and sensitivity information for each of the subproblems are returned, get the remaining target cost estimates for targets that do not yet have an associated cost for each subproblem.
  - Calculate the marginal value, λ, of each target to each subproblem.
  - For each target, compare its marginal value in its current subproblem to its marginal values in other subproblems. If there are other subproblems with higher marginal values, note a potential swap between the target's current subproblem, and the subproblem with the highest marginal value for that target.
  - Rank order, from highest to lowest, the potential gains from swapping targets. Depending on the swapping step size, swap the first n targets that will result in the highest potential gain. Do not swap any targets that appear on the previous subproblem assignment list for targets.
  - Generate new subproblems and send down for detailed planning.
- Send plan out for execution.

5.1.5.1.2.2 Resource Negotiation

Resource negotiation proceeds similarly to target negotiation, however, the assumption here is that target allocations to subproblems are fixed and resource allocations are changeable. Resource negotiation enables another degree of control. It is not difficult to imagine that the initial allocation of resources may not result in the ideal combination of aircraft types. Resource negotiation allows a subproblem to indicate to the master-level planner the degree to which it requires certain types of aircraft in order to prosecute the targets assigned to it. In resource negotiation, subproblems lose their affiliation with air bases, since aircraft in a subproblem can come from any air base.

This resource negotiation is closely aligned with price-coordinated decomposition, wherein the prices of resources are adjusted by the master level and passed down to the sublevel planners. Each sublevel planner passes to the master-level planner plans and sensitivity information identical to that used in target negotiation. Essentially, the master-level planner keeps track of the prices of aircraft in each subproblem. The prices are initialized for all subproblems according to the default pricing (hourly log cost) for aircraft in the scenario definition. Figure 5.1-9 shows a typical matrix of prices for a single subproblem. The price is indexed on the type of aircraft and an air base. Therefore, subproblem SP1's price for J2s from AB1 will be different from SP1's price for J2s from AB2. The combination of a specific aircraft type and air base location is referred to as an aircraft category.



	AB1	AB2	AB3	AB4	AB5
F2W	10	1	10	5	9
J2	11	6	4	20	1
F5E	2	4	1	7	8
F6E	4	5	11	14	14
F7D	7	3	4	3	15
B101	3	1	11	5	12
B102	1	9	14	6	11
B100	2	5	2	8	10

Aircraft Type
Price by Aircraft Type and Airbase for Subproblem *i*

Figure 5.1-9. Prices of Resources for Subproblem i According to Aircraft Type and Originating Air Base.

The prices for each aircraft category in each subproblem are updated individually based on its percentage of resource use, which is reported by the sublevel planner along with the resulting plans. Prices are increased for aircraft categories that are used at a higher percentage, and decreased for aircraft categories used at a lower percentage. Prices were increased by 20% if the aircraft category was used more than 70%, and decreased by 20% if used less than 30%. Prices were kept constant if usage was between 30-70%. By increasing the price of aircraft categories in high demand, the master level attempts to encourage subproblems to use alternative strike packages consisting of lower priced aircraft categories. After several iterations of negotiation, subproblems will make more efficient use of their resources by using high-priced aircraft only for missions that really require them.

Aircraft are swapped to other subproblems based on a marginal benefit calculation similar to the target negotiation case. Here, the marginal gain is calculated for each aircraft category in each subproblem and is given by

$$\lambda = T * P$$

where λ is the marginal value of an aircraft in the category, T is the threshold value (i.e., the lowest ratio of value to cost of the targets in the subproblem plan), and P is the price of the aircraft category in the subproblem. The threshold value is used as a gradient to indicate the gain in the objective value of the subproblem per unit increase of aircraft resources in general. Potential swaps are determined by examining each aircraft category in each subproblem. Swaps are made between the subproblem with the lowest marginal value and the one with the highest marginal value. Once the aircraft category has been decided, the master-level planner still needs to choose specific aircraft to swap. In our experiments, aircraft are chosen to be swapped in order of appearance in the aircraft list as long as they are not currently en route to a target. Once swapping is completed, the modified subproblems are sent to the sublevel planner for detailed planning. New plans cycle up to the master level, which in turn, updates the prices and repeats the process.

Resource Negotiation Algorithm

- Execute the heuristic decomposition algorithm.
- While the maximum number of negotiation iterations has not been reached, continue
  - After plans and sensitivity information for each of the subproblems are returned, update the prices of aircraft categories for each subproblem based on the percentage usage of each category.
  - Calculate the marginal value, λ, of each aircraft category in each subproblem.
  - For each aircraft category, compare its marginal value in its current subproblem to its marginal values in other subproblems. If there are other subproblems with higher marginal values, note a potential swap between the subproblem with the highest and the one with the lowest marginal value for that category.
  - For each swap, choose an aircraft from the subproblem with the lowest marginal value and assign it to its new subproblem. If the lowest marginal value subproblem does not own any aircraft in that category, ignore the swap.
  - Generate new subproblems and send the subproblems for detailed planning.
- Execute plan.

5.1.5.1.2.3 Combined Negotiation

Target and resource negotiation were combined in a variety of configurations in order to take advantage of their complementarity. Combining the two forms of negotiation gives a larger improvement in the global objective value. Figure 5.1-10 shows the three different configurations that were implemented: sequenced, alternating, and parallel negotiation.

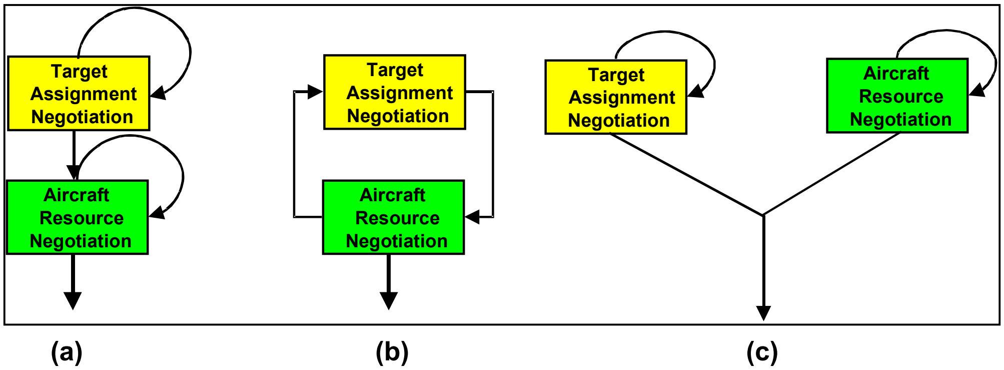

Target Assignment Negotiation
Aircraft Resource Negotiation
Target Assignment Negotiation
Aircraft Resource Negotiation
Target Assignment Negotiation
Aircraft Resource Negotiation
(a)
(b)
(c)

Figure 5.1-10. Negotiation Configurations: (a) Sequential, (b) Alternating, and (c) Parallel.

Sequenced negotiation is structured such that one form of negotiation is completed before another takes place. For instance, the master-level planner starts by executing five rounds of target negotiation while keeping track of the best result. The best allocation generated by target negotiation is then resource negotiated. The overall best result is executed. This allows the subproblems to be "optimized" in terms of one parameter before being optimized for another.

Alternating negotiation performs a single round of each type of negotiation per negotiation cycle. Target negotiation generates subproblems for resource negotiation to modify and vice versa. The best result is saved at each iteration. This allows the master-level planner to trace a zigzag-like pattern through the search space in the target and resource directions.

Finally, parallel negotiation allows both types of negotiation to occur at the same time. Swaps suggested by target negotiation and resource negotiation are executed simultaneously with the best result saved at each step.

5.1.5.1.3 Mission Generation Planner

The Mission Generation planner prioritizes targets, selects the weapons and strike package configuration to be applied to each target, assigns aircraft to targets, and schedules sorties. This planner attempts to maximize target value specified by region and functional category as established via the Commander's Priority Input Matrix. At the same time, it attempts to minimize a total cost function that accounts for attrition risk, operational costs per flight hour, weapon utilization, and reduction in combat effectiveness when pilots are retasked during flight.

The heuristic implementation of the Mission Generation planner serves two functions. First, it provides a baseline against which to compare the performance of the optimal integer programming solution. Second, it provides optimal or near-optimal candidate missions for conditions that the optimal planner specifies. In this second function, the heuristic planner forms the lower level of a decomposition that greatly speeds convergence of the optimal planner.

The heuristic algorithm sequentially assigns aircraft to targets, optimizing the incremental contribution to an objective function for each aircraft mission that the planner generates. This “greedy” approach is suboptimal, but it generates reasonable plans quickly. The heuristic planner uses expected target value divided by expected total mission cost for each mission (combination of a target and the strike package assigned to that target).

The expected target value is determined by the Commander’s Priority Input Matrix and by the probability of inflicting the required damage level on the target. The expected total mission cost is given by

$$\sum_{i \in P} \left(F_i A_i + L_i T_i + R_i \right) + NU$$

where F_i is the flyaway cost for each aircraft i in the package P, A_i is the attrition risk, L_i is the operational cost per flight hour, T_i is the number of flight hours, R_i is a retasking cost, N is the number of weapons to be released, and U is the unit cost for the selected weapon type. The retasking cost is a crude model for the reduction in combat effectiveness if the pilot is retasked during flight and discourages retasking unless there is a good reason to do so.

The Strategic Routing planner supplies the attrition risk and flight time for each individual aircraft sortie. It accounts for the target location, the escort levels corresponding to the strike package configuration, and any strike package assembly and disassembly locations that the Mission Generation planner specifies. The Mission Generation planner uses this information to select the optimal group of aircraft to include in a particular mission and to select the optimal group of missions. If an aircraft is already en route to a target at the beginning of a planning cycle, that aircraft's total cost is assumed to be zero for purposes of selecting the aircraft to apply to that target. (The optimization over missions still considers the actual cost due to attrition risk, flight time, and retasking.) This assumption encourages packages to stay together unless there is a good reason to separate them. Figure 5.1-11 illustrates the cost calculation for an individual aircraft.

Figure 5.1-12 illustrates the procedure for assigning aircraft to a particular mission characterized by a specified start time, target, assembly point location, weapon choice, and package configuration. The planner assigns the required number of each aircraft type (if available) by selecting those with minimum total cost. In the example in Figure 5.1-12, the package configuration requires two aircraft of type F2W (of which six are available), two of type F5E (of which four are available), and one of type J2 (of which two are available). Two aircraft of type B101 are also available, but the package configuration does not require any of those. (In practice, the planner does not evaluate

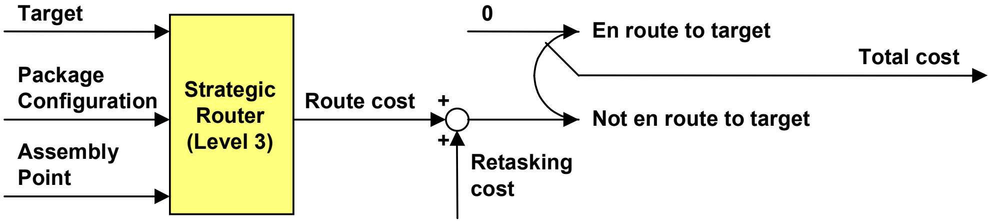
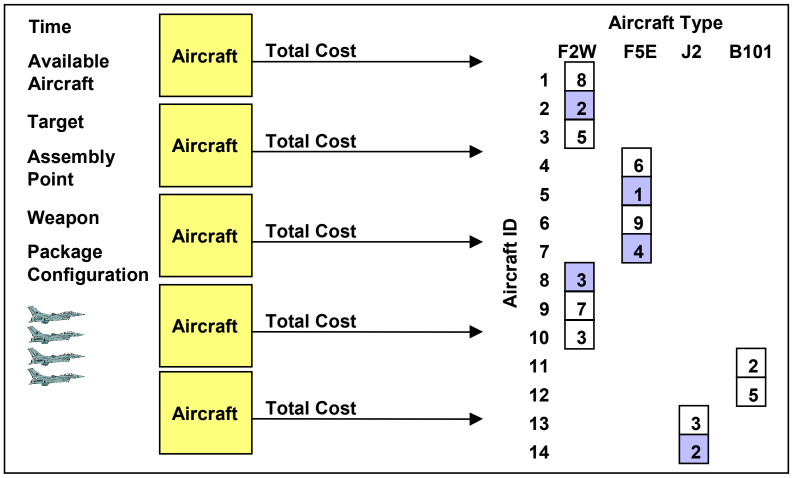


Figure 5.1-11. Aircraft Cost Calculations.



Aircraft ID	F2W	F5E	J2	B101
1	8			
2	2			
3	5			
4		6		
5		1		
6		9		
7		4		
8	3			
9	7			
10	3			
11				2
12				5
13			3	
14			2	

Figure 5.1-12. Package Formation.

costs for aircraft types not required by the package configuration.) The numbers in the squares to the right of the figure represent the cost for each aircraft, and the colored squares indicate the (minimum-cost) aircraft that the planner chooses under these conditions.

The previous discussion shows how the planner assigns aircraft and evaluates the cost for a specific package configuration. The planner repeats this process for all candidate package configurations and selects the one that maximizes the objective function (value divided by cost). The planner chooses package configurations from a prespecified list of alternatives. Only those package configurations that have enough strikers to deliver the required number and type of weapons to the target are valid candidates for a particular mission. In selecting the optimal package configuration, the planner effectively chooses the jammer and Suppression of Enemy Air Defenses (SEAD) escorts to minimize the expected total cost for the mission. While escorts incur operational costs, they reduce the expected attrition. Hence, the optimization encourages the use of escorts primarily for high-risk missions. Figure 5.1-13 illustrates the procedure. Note that the planner must now consider the complete objective function (both value and cost) because different escort levels result in different routes. Hence, the package configuration affects the time on target, which in turn affects expected value for time-sensitive targets.

The choice of weapon for a particular target determines the candidate package configurations that are valid for that target. Some targets may have multiple weapon types that are effective against them. The planner selects the weapon type that maximizes the objective function. The weapon type selection affects the expected value via the expected damage level against the target and the time on target for the optimal package that can carry that weapon. The weapon type selection affects the expected cost via the unit cost per weapon and the number of weapons required to achieve the threshold damage level. Figure 5.1-14 illustrates these factors.

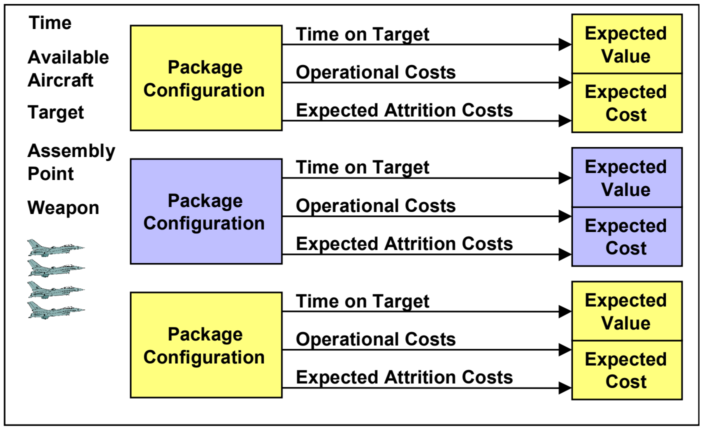


Figure 5.1-13. Package Configuration.

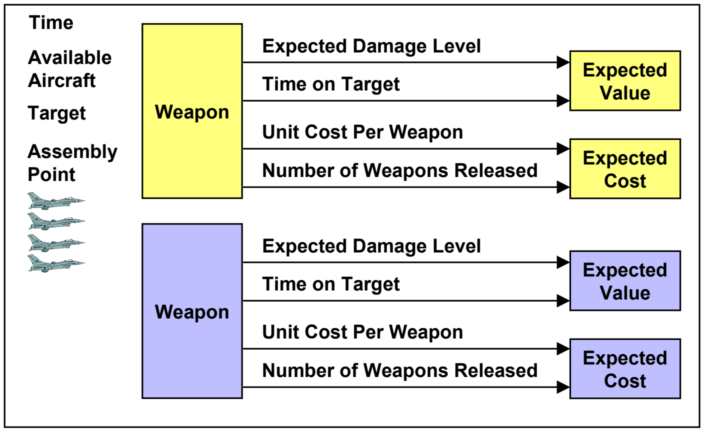


Figure 5.1-14. Weaponeering.

The optimal package selection depends on whether the aircraft in the package plan their routes directly from the air base to the target or whether they plan first to assemble at a point outside the threat area. In the first case, escorts do not provide any protection for the strikers unless they come from the same air base. In the second case, the strikers are protected during the route segments between the assembly point and the target location. A good assembly point location is one that is essentially "on the way" between the air bases and the target location and that is positioned such that there is little risk to the strikers flying between the air base and the assembly point. The planner chooses the assembly point from a list of candidates to maximize the objective function, as Figure 5.1-15 illustrates. In addition to the candidate assembly points, the planner always considers the option not to use any assembly point at all. The planner synchronizes the time on target for all aircraft in a package. This means that if aircraft from multiple bases form a package, the aircraft may take off at different times. The aircraft in a package arrive at any assembly point at the same time and take the same route between the assembly point and the target. Analogous conditions hold for aircraft leaving a target area, possibly passing through a disassembly point outside the threat area, and returning to base. The heuristic implementation simplifies the optimization problem by always using the same point for both assembly and disassembly.

The procedures described above derive the optimal parameters for a mission to prosecute a particular target given the available aircraft at a specific time. In the process of doing this, the planner also evaluates the objective function corresponding to the optimal mission for that target. The heuristic implementation of the planner prioritizes the targets in a “greedy” manner. This means that for a given time and set of available aircraft, the planner always selects the target with maximum objective function value to be the next one prosecuted. The first panel of Figure 5.1-16 illustrates this point. The numbers in the boxes labeled E{V}/E{C} correspond to the objective function for

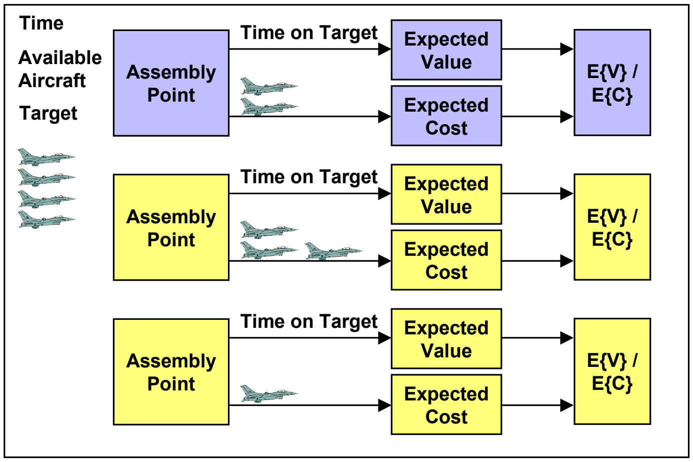


Figure 5.1-15. Assembly Point Selection.

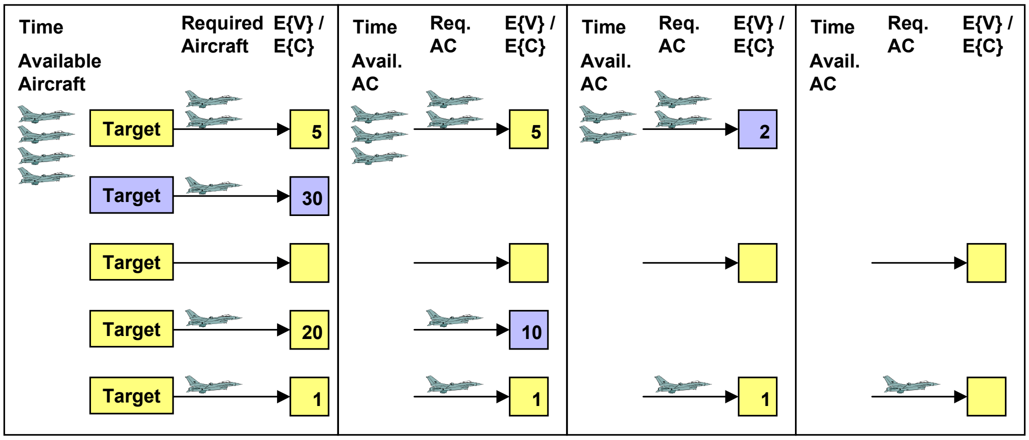


Figure 5.1-16. Wave Planning.

each target, and the planner selects the target with maximum objective function (30). (Some targets may be infeasible with the available aircraft at the specified time; these cases are represented by blanks in the corresponding objective function boxes.) The planner evaluates the objective function for each target assuming that all available aircraft are at its disposal. After selecting a target with its corresponding mission, the planner removes that mission's aircraft from consideration and repeats the process to determine another target to prosecute at the same time. The second and subsequent panels of Figure 5.1-16 illustrate this process. Note that the objective function may decrease because some of the aircraft are no longer available. The planner schedules an entire "wave" of sorties in this manner, continuing until either the available aircraft or targets are exhausted.

The planner generates a schedule of sorties for an entire planning horizon in waves using the procedure described above. The waves are asynchronous in that the planner does not wait for all aircraft in a wave to return to base before planning the next wave. Instead, the planner sends a wave whenever enough aircraft are available to form at least one new mission. The planner keeps trying to form new waves every time an aircraft returns to base. This process continues until the end of a specified planning horizon. Any missions that begin during the current planning horizon are valid, so there is typically activity beyond the end of the planning horizon in the plan. Figure 5.1-17 illustrates the process of repeatedly calling the "wave planner" from Figure 5.1-16 during a planning horizon.

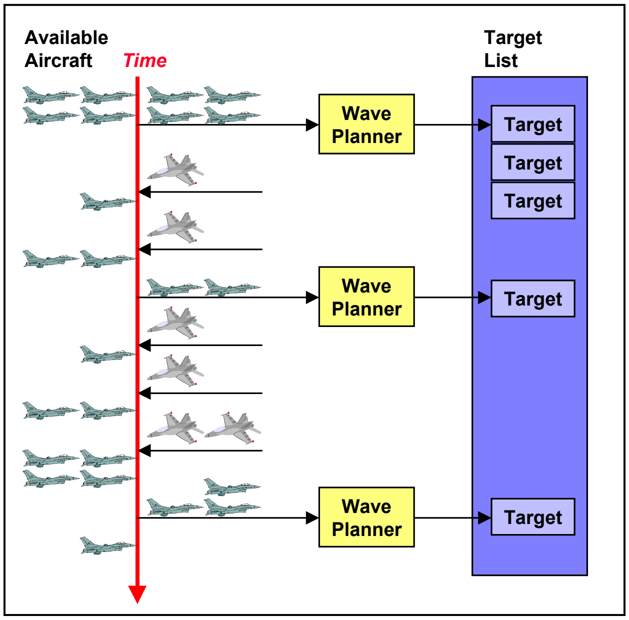


Figure 5.1-17. Greedy Heuristic.

In summary, we have described the heuristic algorithm by which the Mission Generation planner prioritizes targets, selects the weapons and strike package configuration to be applied to each target, assigns aircraft to targets, and schedules sorties. In performing these functions, the Mission Generation planner depends heavily on the Strategic Routing planner to generate aircraft routes that represent the optimal tradeoff between attrition risk and operational costs.

5.1.5.1.4 Router

The Level-3 planner is a strategic router that supports the Mission Generation planner by providing the cost of constrained minimum risk routes for specified aircraft target pairs. In particular, the route-planning problem may be posed as follows.

Given:

- **Mission parameters** including:
  - Start location (base or en route) and return base.
  - Required ingress and/or egress assembly points.
  - Target location.
  - Set of all tanker locations.
- **Aircraft parameters** including:
  - Fuel endurance.
  - Pilot endurance.
- **Threat model parameters** including:
  - Threat density.
  - Detection range.
  - Likelihood of engaging.
  - Pk.

and

- **Escort level** representing one of:
  - No escort.
  - Wild weasels.
  - Weasels plus escort jammers.

Determine:

- **A strategic-level route** (10- to 30-km grid spacing for waypoints) that is:
  - Fuel feasible.
  - Minimizes risk from threat engagement at 10- to 30-km resolution.
  - Allows for up to two refueling activities on each segment.
  - Adheres to specified ingress and egress assembly points.

It should be noted that the underlying threat model assumes that threats locations are not known in detail. Although ISR may indicate discrete threat locations, the information source is likely to be aged by at least 24 to 48 h by the time the mission is executed. Hence, one may infer that mission planning should rely on a probability density model of threat locations rather than on precise threat location information. The probability density has units of threats per kilometer squared averaged across a grid box that is currently 30 km on a side. The purpose of the Level-3 router is to generate a route as described by waypoint segments that minimizes potential exposure to this representation of threats. The resulting routes attempt to avoid passing through areas containing high densities of threat systems subject to the constraint that the cost of time as well as fuel endurance constraints precludes circuitous paths that travel far and wide to avoid threats. It should also be noted that since the threat model includes the effects of defense suppression escorts such as HARM-shooting weasels and radar jammers, there may be different routes generated for different package compositions between otherwise identical mission constraints.

A C-code implementation of an A* search-based planner has been integrated with the Mission Generation planner. The route structure is shown in Figure 5.1-18.

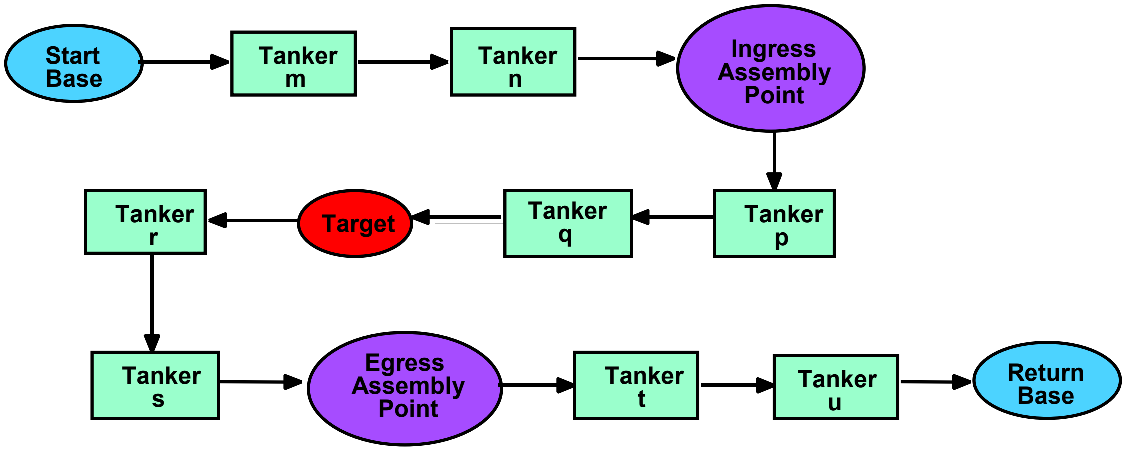


Figure 5.1-18. Possible Refueling Options for Nominal Mission Structure.

The boundaries of the A* search grid superimposed on the Cyberland theater geometry are shown below in Figure 5.1-19.

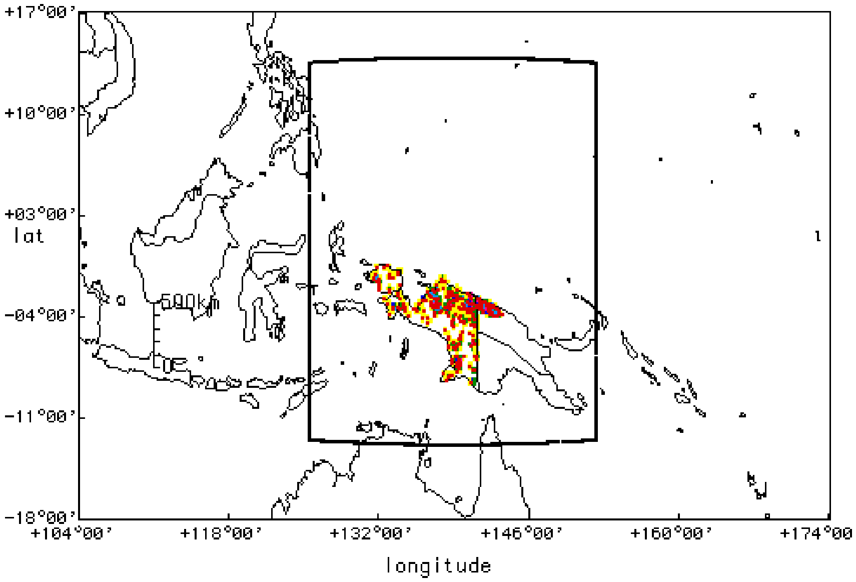


Figure 5.1-19. Boundaries of Search Grid in Cyberland Scenario.

Figure 5.1-20 illustrates the results from the router for a 30-km gridding with a required assembly point on ingress. The legend indicates base, tanker, target, and assembly point locations.

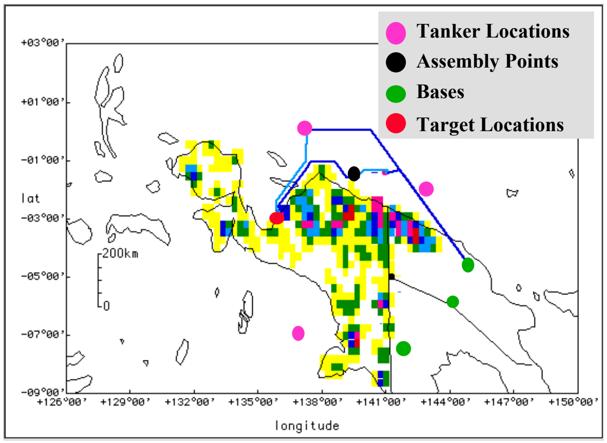


Figure 5.1-20. Example Routing with Required Assembly Point on Ingress.

Given the underlying threat model, the router was implemented to produce fuel-feasible minimum attrition cost routings according to a mission structure, which included:

- A vertex corresponding to take off from a base or an arbitrary en route start location.
- A possible vertex for an ingress assembly point.
- A vertex at the target weapon release location.
- A vertex at a possible egress assembly point.
- A vertex at the specified return base.

Additionally, up to two aerial refueling tanker segments could be added between each of the other segments, as necessary. Each segment is constructed as a minimum cost route and the full combinatorial enumeration of tanker possibilities is explicitly evaluated with pruning of those cases that are illogical. Examples of the latter include doubling back to a previous tanker rendezvous before the run-in to the target. The full enumeration may involve several thousand possibilities, but the precomputation of the min-cost search route from the limited number of specified locations (e.g., bases, tanker locations, assembly points) renders each enumeration to involve a table lookup involving no more than milliseconds of compute time. Hence, after an initialization time that may range up to about 15 s, fuel-feasible min-cost routings at 30-km resolution with explicit selection of tanker refuelings takes only tens to hundreds of milliseconds per invocation.

The A* search generates a nodal cost map onto an underlying network grid that is used for the A* search. The attrition cost model is described in detail in Section 6. Although the min-cost route is independent of the scaling of the nodal cost numbers, the inclusion of the cost of time leads to the possibility of more direct routing through high threat areas as a result of increasing the level of defense suppression escort service.

In addition to generating the min-cost route, including the planned arrival times at all waypoints, the router reports the estimated attrition risk along the determined path. This result is used in the Level-2 heuristic to enforce the risk management constraint. Candidate missions that exceed a risk level that is specified as a function of aircraft type are rejected by the heuristic.

Additional functionality was added to address the case of targets that relocate over time and threats that may change with time because of relocation or attrition. For the former case, the simplest mission repair is to shift the vertex of the mission command that contains the target, holding the remainder of the plan constant. The limited mobility of ground equipment during the interval between planning cycles and during the mission duration suggests this as a possible correction. Given the current, en route aircraft location and the updated target location, the router evaluates the difference in mission time and estimated attrition risk that results from this simple correction and updates the remainder of the waypoint arrival times to reflect the change. If the event-based controller rejects the change as being excessively costly, then a new route for this mission is generated by A* search given updated locations.

For the case of time-varying threat parameters, the router evaluates the difference in mission risk between the original set of parameters on which the route was based and the updated or current set of threat parameters. The event-based controller uses this information to decide when to initiate a new A* search to replan the route.

Finally, it should be pointed out that mission route structure addresses a given release point for a single target. There are a couple of problems that are not yet addressed in the current JFACC Air Operations Controller:

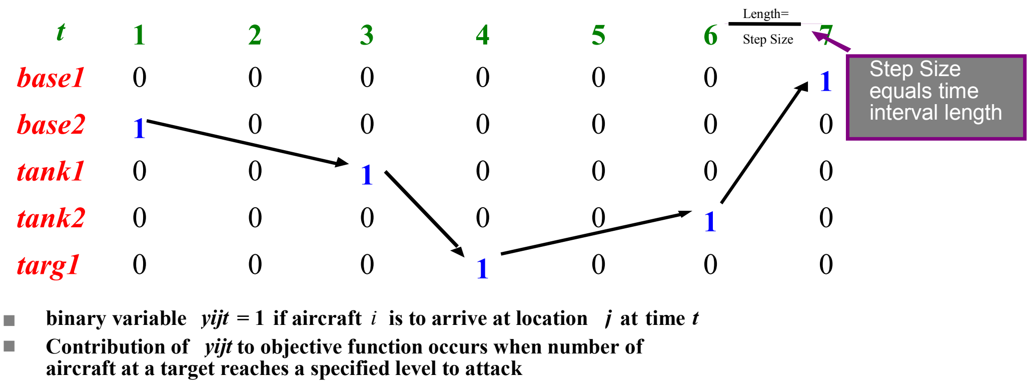
- Selection of release point for weapons with significant standoff range.
- Routing for a heavy bomber tour that visits a number of targets.

A number of algorithms were constructed to address these issues, but the results were not developed or integrated in the experimental development context of the current work scope. Both of these are issues would require further development for a go-to-war implementation.

5.1.5.2 Optimal Approach for Mission Generation Planner

We developed two techniques for generating missions at the mission generation level. Above, we covered the heuristic approach; in this section, we cover the application of optimization techniques to generate missions. We developed optimization techniques to generate missions because we felt these techniques were capable of generating more effective plans than a heuristic or human could.

Early efforts at developing an optimal formulation of the Mission Generation level resulted in the main decision variable representing the location of aircraft at every period of time. Figure 5.1-21 is a representation of this decision variable.



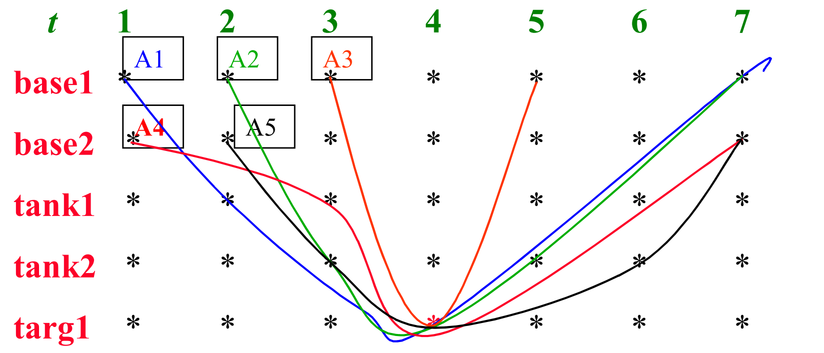
t	1	2	3	4	5	6	7
base1	0	0	0	0	0	0	1
base2	1	0	0	0	0	0	0
tank1	0	0	1	0	0	0	0
tank2	0	0	0	0	0	1	0
targ1	0	0	0	1	0	0	0

- **binary variable *yijt* = 1 if aircraft *i* is to arrive at location *j* at time *t***
- **Contribution of *yijt* to objective function occurs when number of aircraft at a target reaches a specified level to attack**

Figure 5.1-21. Aircraft Time Period Variable.

This decision variable provides flexibility because a solution to these variables essentially solves the entire air operations problem while easily allowing for features such as multiple target sorties. However, applying constraints to this variable makes the model unsolvable in any realistic time frame. Simultaneously enforcing constraints such as aircraft performance, weapons capabilities, and fuel constraints requires many complex constraints. Specifically, representation of aircraft performance constraints of speed and distance flown requires many constraints on the aircraft time period variables. The constraint is enforced for every aircraft, pairing of locations, and time period. Therefore, with a realistic example of 100 aircraft, 505 locations (500 targets, 5 bases), and 16 time periods (4 h of 15-min intervals), there would be approximately 400 million timing constraints. An integer program with these constraints is intractable to solve, motivating a composite variable model.

The composite variable model produced encouraging results. Figure 5.1-22 illustrates the composite mission variable.



- **A Mission variable is an aggregate decision variable defined by:**
  1. Target to be attacked
  2. Package of aircraft (including escorts) and weapons
  3. The bases from which the aircraft come
  4. Routes, including timing, for aircraft in the mission

Figure 5.1-22. Composite Mission Variable.

The composite variable approach produces an equivalent but stronger integer programming formulation through the translation of decision variables. The composite mission variable allows for a model with a much smaller number of constraints, and the linear programming (LP) solution to the Mission Variable Model is close to integer. However, as one can imagine, there are many possible combinations of targets, package configurations, weapons, assembly points, and routes to form missions. Therefore, while the composite mission variable reduces the number of constraints and the complexity of the model, the number of columns in the optimization model is very large.

Fortunately, we have techniques to "sift" through the large number of missions and only consider the best. Before we present the techniques, the composite mission variable integer programming model is presented.

5.1.5.2.1 Mission Variable Model

Decision Variables

$x_{mt} \in \{0,1\}$ - 1 if mission m starts at time t

Indices

m = Mission. Mission defines the target, contains the type of aircraft, including escorts, defines the routes and assembly points, and defines the bases that provide the aircraft

t ∈ $\{1,2,\ldots,T\}$ - time in periods, T being last time period

b = Base

a = Aircraft type

r = Target

c = Committed aircraft number

Sets

MC = Set of all missions that have at least one committed aircraft

M(r) = Set of missions that prosecute target

Data/Parameters

V_{mt} = Expected value of starting mission *m* at time *t*. The mission defines the target, and the value is representative of the time the aircraft arrive over the target

F_a = Fly-away cost of aircraft type *a*. Impact of losing aircraft of type *a* on the future missions

CV = Commander tuning parameter for relating attrition cost to target value

R_{abm} = Probability of losing aircraft of type *a* from base *b* in mission *m*

Q_{abm} = Quality of aircraft of type *a* from base *b* in mission *m*

AC_m = Attrition cost that is a function of F_a, R_{abm}, Q_{abm}, and *CV*. $AC_m = \sum_{ab} F_a * Q_{abm} * R_{abm} / CV$

MT_{abm} = Mission time for aircraft *a* from base *b* in mission *m*

S_{abt} = Supply of aircraft type *a* at base *b* at time *t*

CD_{mc} = Binary parameter, which is 1 if mission *m* uses committed aircraft *c*

Objective Function

Maximize the expected value of the targets attacked less the attrition cost. Maximize the sum of the target value minus the attrition cost of all missions executed over all time.

$$\underset{x}{Max} \sum_{t} \sum_{m} (V_{mt} - AC_m) x_{mt}$$

Constraints

Aircraft Supply: Missions executed are constrained by available aircraft. For all aircraft type, base, and time period combinations, the sum of all the missions using that combination must be less than the supply of that combination.

$$\sum_{m} \sum_{t' \in [t - MT_{amb}, t]} Q_{abm} * x_{mt'} \le S_{abt}, \forall a, b, t$$

Hit Once: Every target can be planned to be hit a maximum of one time during the plan horizon. For all targets, the sum of all missions over time attacking that target must be less than or equal to one.

$$\sum_{m \in M(r)} \sum_{t} x_{mt} \le 1, \forall r$$

Committed Aircraft: Aircraft committed on the ground or in the air at the beginning of the plan horizon must be included in one and only one mission at the beginning of the new plan horizon. After completing this mission, these aircraft are no longer committed and therefore can be used for more missions. For all committed aircraft, at time period zero, the sum of all missions using that committed aircraft must equal 1.

$$\sum_{m} CD_{mc} * x_{m0} = 1, \forall c$$

Committed Aircraft 2: Missions using committed aircraft cannot start after time zero because the mission is no longer defined. For all missions that have committed aircraft, and for all time after time zero of the plan horizon, the mission cannot be executed.

$$x_{mt} \le 0, \forall m \in MC, t \in [1, T]$$

5.1.5.2.2 Mission Variable Generation

In order to solve the Mission Variable Model, the mission variables must be created. Just as there are constraints on the mission variables that can be in the solution, there are constraints on the missions created for the Mission Variable Model. In both the heuristic and optimal controller, we assume each mission only attacks one target. The following constraints are imposed on the missions created for each individual target.

Mission Variable Constraints

Weaponeering: We assume each target has weaponeering options that are created before the controllers generate plans. Each weaponeering option specifies the type and quantity of bombs that can be dropped on the target in order to achieve a specified damage level.

- Missions can only carry prespecified weaponeering options.

Package Configuration: We assume commanders establish allowable package configurations. A package configuration specifies the type and quantity of aircraft that can be used to attack a target. In addition, aircraft can only carry a single weapon configuration at a time. Obviously, controllers should assign package configurations to a target that are capable of carrying one of the prespecified weaponeering options.

- Missions can only have aircraft packages that are one of the prespecified package configurations, and missions should not have package configurations that cannot carry one of the prespecified weaponeering options.

Routing: The router, as shown in Section 5.1.4.1, generates optimal routes based on risk and time costs.

- Missions created can follow only the routes that are created by the router.

  - The router constrains aircraft to refuel only at prespecified tanker locations.
  - The router constraints aircraft to their range capabilities.

Permanent Aircraft Supply: The Mission Variable Model has resource supply constraints so that the optimal plan solution does not use more aircraft than available. For example, if 20 F-15s are located at a base, then 10 missions that each required 2 F-15s would be the maximum number to simultaneously execute. However, if a base only has one F-15 assigned to it, it would not be reasonable to generate any missions that required more than one F-15 from that base. We will call the supply of aircraft at a base when all aircraft are on the ground the Permanent Aircraft Supply.

- Missions created cannot use more of an aircraft type from a base than the Permanent Aircraft Supply of that aircraft type at that base.

Assembly Points: We assume allowable assembly point locations are prespecified by Commanders. Assembly points are used for multiple-base packages with escorts to assemble with their escorts before entering the threat area.

- Missions created can only allow aircraft to assemble at prespecified assembly locations.

Price-Coordinated Mission Generation

As one can imagine, the number of missions and columns of the Mission Variable Model grows exponentially with the scenario size. The following is an example of how to calculate an upper bound on the number of possible missions for a scenario:

Number of Missions Example

- The number of missions is equal to all the ways to combine the targets, assembly points, package configurations, weaponeering options, and base/aircraft combinations.
- Scenario:
  - 1000 targets.
  - 3 assembly points.
  - 25 package configurations.
  - 3 weaponeering options per target.
  - 5 bases.
  - 5 aircraft types.
  - 6 aircraft of each type at each base for a total of 150 (6*5*5) aircraft.
  - 4-h time horizon with 15-min intervals = 16 time periods.
- Upper bound assumptions for example scenario.
  - Assume each package configuration requires 3 aircraft.
  - Assume all aircraft are in range of the targets.
  - Assume all assembly points to target combinations are in range of aircraft.
  - Each base has 4 options for the number of aircraft in the package it provides because there are exactly 3 aircraft per package and each base can supply 0, 1, 2, or 3 of the aircraft. This adds up to 4^5 combinations. This is certainly an upper bound, but an easy simplification of the combination of aircraft calculation.
- Number of missions = $1000 * 3 * 25 * 3 * 4^5$ = 230 million missions.
- Number of columns = 230 * 16 time periods = 3.7 billion columns.

With 3.7 billion columns, it would take much more computer memory and solution time than available for the Mission Variable Model to solve. For most realistic scenarios, this is a very high bound because of the high bound on the aircraft combinations, assuming 3 aircraft per package configuration, assuming all aircraft types are located at all the bases, and assuming all aircraft are within range of all targets. In addition, many missions would be clearly better candidates over others so that a simple filtering scheme could be used to decrease the mission set. However, for this scenario example, even developing a more realistic bound and then filtering many of the obviously poor missions out of the mission set would leave hundreds of thousands of realistic candidate missions. This still corresponds to millions of columns in the Mission Variable Model.

This large set of missions required provides motivation to decompose the problem. We generate an initial set of missions for the Mission Variable Model to choose from, using the heuristic or via other means. We then solve the Mission Variable Model using this initial mission set. The reduced cost $\overline{C}_{mt}$ of each x_{mt} variable, including those representing missions that are not currently in the mission set can be calculated

$$\overline{C}_{mt} = V_m - \sum_{ab} (F_a * Q_{abm} * R_{abm} / CV) * y_{ab} - \sum_{ab} MT_{abm} * P_{ab} * y_{ab} - \sum_{c} P_c * y_c - P_r$$

Here, y_{ab} represents the quantity of aircraft of type *a* from base *b*. P_{ab}, P_c, and P_r are the dual prices of the aircraft supply constraints, committed aircraft constraints and hit-once constraints, respectively. The reduced costs of the mission variables are equal to the objective coefficients of the mission variables in the Mission Variable Model minus the dual prices of the constraints. Missions that maximize the reduced cost equation are the missions whose addition to the mission set would most increase the objective value of the optimal solution to the linear programming relaxation of the Mission Variable Model. These missions also tend to increase the objective value of the integral Mission Variable Model, thanks to the strength of the Mission Variable Model linear relaxation.

The decomposed model solution process adds the missions with greatest reduced cost to the mission set, resolves the Mission Variable Model, and repeats until either all reduced costs are negative or, optionally, until some other stopping criterion has been met. At each iteration, the reduced cost equations change as a result of changes in the dual prices in the solution to the Mission Variable Model. Negative reduced costs correspond to mission variables that will not be selected in the optimal solution to the LP relaxation. Thus, stopping when all reduced costs are negative produces the optimal solution to the LP relaxation of the Mission Variable Model. As a final step, branch and bound is used to produce an integral solution to the Mission Variable Model.

It is likely that some missions that would be included in the optimal integral solution to the Mission Variable Model are not added to the mission set, resulting in suboptimal integral solutions. Fortunately, the LP relaxation of the Mission Variable Model is relatively strong. Not only is an integral solution found rapidly during branch and bound, but the integral solution objective value is very close to the objective value of the LP relaxation (which is an upper bound on the integer program (IP) objective value). For several scenarios with adequately small mission sets, we enumerated the entire mission set and solved the Mission Variable Model to optimality. We compared this solution to the solution found through the price-coordinated decomposition, and found that all objective values produced through the decomposition were within 0.3% of the fully optimal objective value.

The subproblem optimizer employed here relies on an enumerative technique to generate mission variables with maximal reduced cost for each target at each iteration. At each iteration, if the dual prices have changed, the subproblem optimizer generates the new set of maximally reduced-cost candidate mission variables that are used by the Mission Variable Model. The subproblem optimizer consists of the same mission generation code employed inside the heuristic planner. Whereas the heuristic planner calls this code to generate missions and selects a feasible set of them, the mission variable generator encodes every maximal reduced-cost mission as a mission variable for the Mission Variable Model to use in optimization. Whereas the heuristic can take considerable time to generate missions in its search, the mission variable generator's exhaustive search over all the ways to attack targets is fast since it considers only one target at a time.

The solution process is as follows:

1. Initialize P_{ab}, P_c, P_r to zero. Set Z_0 (last objective function solution) equal to zero.
2. Call mission variable generation for each target with the current values of P_{ab}, P_c, P_r. Add the missions to the mission set.
3. Solve the LP solution to the Mission Variable Model with the current mission set. Set Z_1 (current objective function solution) equal to LP objective function.
4. If no new missions were generated (because reduced costs did not change adequately), or alternatively, if $Z_1 < Z_0$ + (small %)* Z_0 go to Step 5. Else, set Z_0 equal to $Z_{1,}$ update P_{ab}, P_c, P_r using the dual prices of the LP solution, and return to Step 2.
5. Solve IP solution to Mission Variable Model. This solution is the plan for the current planning cycle.

   This algorithm typically takes 4-7 iterations to solve. Table 5.1-2 shows an example run of the algorithm for a scenario with 910 targets.

Table 5.1-2. 910-Target Scenario.

Iteration	Objective Value	Num X>0	Num a/b Sol	Num X >0 for Iter	Num Missions	Num Duplicates per Iteration
0	0	0	0	0	0	0
1	1.6262	37	2	37	415	0
2	2.6884	57	3	27	744	7
3	2.978	62	4	16	1067	52
4	3.1006	63	3	8	1270	96
5	3.1059	65	3	5	1373	209
6	3.0399	62	0	0	1373	0

The 3rd column is the number of missions assigned, the 4th column is the number of fractional solutions in the LP solution, the 5th column is the number of missions used from the new missions generated on the current iteration, the 6th column is the total number of missions in the mission set, and the last column is how many suggested missions were duplicates of previously suggested missions. The last iteration is the integer programming solution; this explains the drop in objective value and number of missions in the solution compared with the previous LP solution. This example is typical, the objective value increases dramatically on the first few iterations as many new missions are added, and then the number of good missions to add to the mission set decreases along with the decreasing change in the objective value. Notice that the number of fractional solutions is only 2 to 4. This is a beneficial feature of the model; most iterations have low numbers of fractional solutions in the LP solution that allows for fast branch and bounding to the integer solution.

We use a commercial solver, Xpress-MP, to solve the Mission Variable Model. The optimization code is written in C/C++ and Xpress-MP libraries are called in the code. The performance of this algorithm is impressive; it can quickly generate effective plans for large scenarios. Performance results will be presented in the experiments section.

5.2 JFACC Architecture

A hierarchical architecture of subcontrollers that mirrors the decomposed JFACC problem has been developed and applied to our family of JFACC scenarios. The architecture is composed of basic building blocks (subcontrollers), each incorporating the functions of "observe-orient-decide-act," and which are termed in this report as "monitor-diagnose-plan-execute." At the higher levels, planning time frames are long, objectives are broad, and decisions are aggregated. At lower levels, the reverse is true.

This controller structure was chosen to take advantage of its good properties:

5. ***Problem Tractability:*** Decomposing the problem into stratified levels over space, as well as time, is often the only way a problem can be solved.
6. ***Conformance to Human Organizations:*** Quite often, new decision-making techniques need to operate within an existing human organization.
7. ***Reduction in Computation Time*:** Hierarchical decomposition should always reduce the time required to solve a problem from that needed to solve a single, monolithic formulation of the problem. Even when single formulations are solvable, time constraints on execution may argue for the time savings that can be obtained through decomposition.
8. ***Scalability:*** Hierarchical structures are extraordinarily scalable, as evident from their wide application in society, from small companies to large military organizations.

The main penalty paid for employing a hierarchical approach is that design and implementation of a system of controllers can be more involved than that of a single controller node. The subcontrollers in the hierarchy must be designed to be highly cooperative and interactive, accepting compatible objectives and constraints from above, and listening for problems and opportunities from below. Also, the orchestration of ongoing activity within the hierarchy is more demanding when a number of subelements need to work together. But the implementation of this orchestration is a one-time cost that can be spread over multiple applications once it is developed.

This section describes the design and implementation of the JFACC hierarchical controller architecture, as applied to the strike planning problem described in Section 3. Since the manner in which the control problem is decomposed drives the controller architecture, it will be discussed first. Then the architecture itself will be described, followed by its concept of its operation.

5.2.1 Decomposition

The decomposition of large systems into hierarchies can be guided in different ways. Among them are included:

1. ***Existing Human Organization*:** Most complex systems are already controlled by existing human organizations that, to a large degree, are hierarchical. New systems and technology that are to enhance the operation of the existing system must somehow fit gracefully within the prevailing operations and procedures. Often, the best way to do this is to conform wholly, or at least partially, with the structure of the existing organization.
2. ***Hierarchical Mathematical Techniques*:** The system's mathematical model can yield novel decompositions, with possible improvements in performance. These decompositions derive from the natural fault lines within the structure of the problem and strive to find organizational elements that are as independent as possible. A purely mathematical approach to decomposition, which is implemented in software, can also dynamically adapt to the current situation more freely that a purely human organization.
3. ***Computational Limitations*:** Limits on computation, or manpower, can impact the responsibilities (workload) allocated to each organizational element. This, in turn, determines the span of control (number of elements) at each level in the hierarchy.

Computational limitations tend not to be an issue with today's powerful computers, and the efficiencies that can be achieved with formal techniques are often overwhelmed by the advantages/requirements of conforming to the prevailing structure of existing organization. This project based its hierarchical controller on the current way in which resources are allocated and directed, but not on the 72-h, rolling horizon process that is currently in place. (Formal techniques were also examined during the course of the project, however the reported experimental results were obtained with the decomposition described in this section.)

The Existing Organization

The military organization for determining, prioritizing, allocating, and prosecuting targets falls naturally into a hierarchical structure based on people, resources (aircraft, weapons, etc.), and targets:

- ***Theater*****:** The theater Commander or his delegate has responsibility for the conduct of all operations in the theater, including the allocation of target lists to subordinate air wings.
- ***Air Wing*****:** The air wing Commander has access to a pool of aircraft, weapons, and supporting resources to carry out the objectives set by the theater Commander. He organizes his resources into strike packages, which work together to accomplish a mission objective. The different aircraft in a strike package perform specialized functions that contribute to the overall effectiveness of the package. Fighters in a strike package, for instance, engage enemy air patrols that could hamper a bombers' ability to reach their targets.
- ***Strike Packages*****:** The strike commander is responsible for carrying out his objective and sets the objectives for individual pilots that support the overall objective. These subobjectives will involve routing choices, timing, escorting assignments, assembly points, and other low-level decisions critical to the mission's success.

Planning Specification

A given planning function is characterized by its objective, the resources allocated to it, the time horizon over which it is to plan, and the decisions it is to make. These attributes, for the planner types need at each level in the hierarchy, are specified in Table 5.2-1.

Table 5.2-1. Planner Decomposition Specification.

	Level 1 - Theater	**Level 2 - Air base**	**Level 3 - Package**
Objective	• Minimize time horizon over which all targets are prosecuted • Maximize rewards/costs through negotiation with Air Base Planner	• Maximize rewards/costs: • Rewards: Accumulated target value • Costs: Weapon usage, attrition risk, fuel usage, retasking cost	• Maximize rewards/costs: • Rewards: Accumulated target value • Costs: Weapon usage, attrition risk, fuel usage, retasking cost
Resources	All aircraft	Allocated air base aircraft	Allocated package aircraft
Horizon	5 days	4 h	2 h
Decisions	Allocation of target sets to air bases	• Selection of aircraft package configuration • Allocation of targets to packages	Routing of aircraft to targets

5.2.2 Architecture

The basic controller architecture can be specified easily, once the hard work of decomposing the planning problem has been done and the individual planning functions within the decomposition have been defined. This subsection specifies the pieces of the architecture and their relationship to one another. How they work together to achieve an objective is discussed in the next subsection.

5.2.2.1 Components and Structure

A simplified three-level hierarchy, commensurate with the decomposition of the previous section, is shown in Figure 5.2-1. A full "four-box" controller is placed at each point in the hierarchical decomposition, and they relate to exactly one parent and several children in the tree. There are no peer-to-peer relationships in a hierarchy. Space limits prevent all five air wings and the numerous packages from being shown.

Actually, the number of package controllers and, for that matter, possibly air wing controllers will dynamically vary over time, depending on current requirements and how the hierarchy is managed. Typically, there will only be as many package controllers as there are current defined packages. Notional packages that are planned for tomorrow or next week need not be in the controller hierarchy now. They can be added as they are needed. With a little flexibility in the human organization allowed, even air wings can conceivably be reconfigured dynamically to have more or fewer than the five we have specified in the scenario. Again, it all depends on requirements and the allowed flexibility in restructuring the controller.

5.2.2.2 Meta-Controller

An important part of the architecture not mentioned so far is the Meta-Controller. This "controller's controller" is single four-box controller, and is responsible for monitoring and controlling the controller hierarchy of Figure 5.2-1.

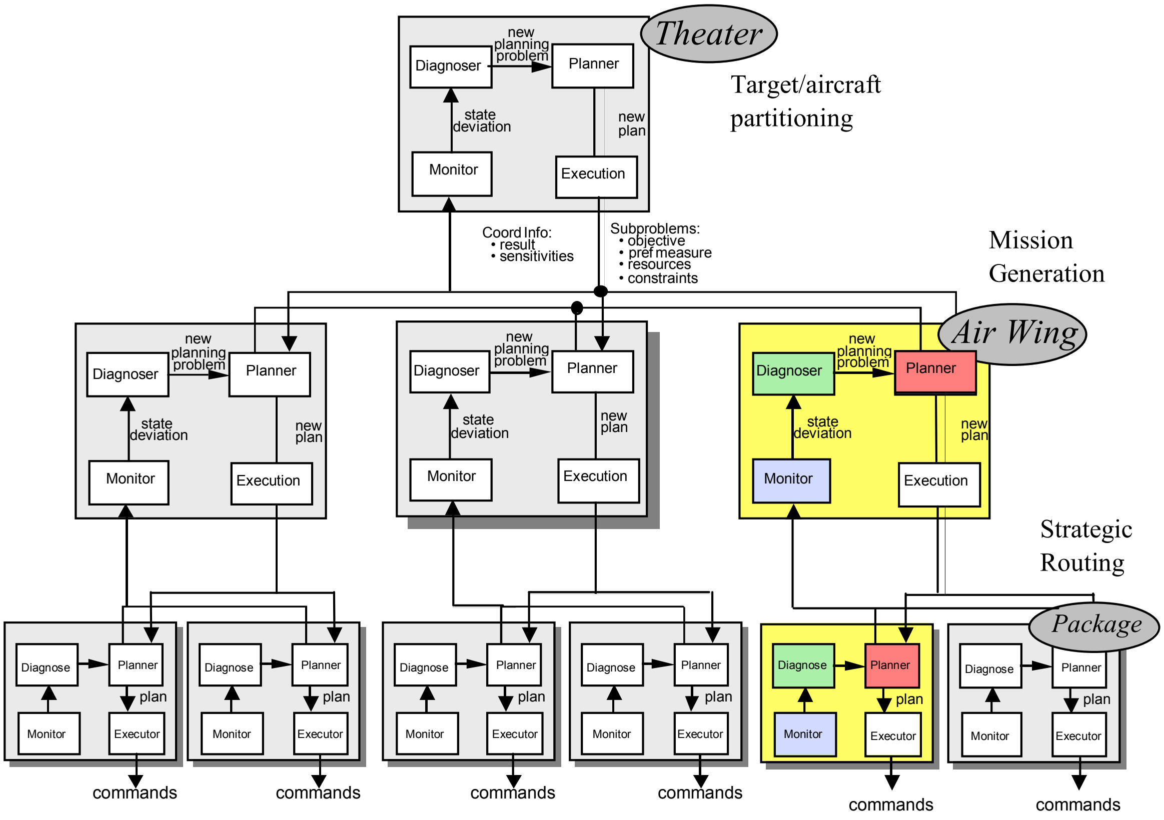


Figure 5.2-1. State Feedback in a Hierarchical Controller.

Where the Controller Hierarchy's domain is some piece of the real world (e.g., the JFACC world), the Meta-Controller's domain is the Controller Hierarchy itself, which is a reasonably complex system in its own right.

The Controller Hierarchy could be burdened with managing its own internal activities, as well as the activity in the real world. But this can create difficulties and add to its own complexity. A Meta-Controller has the advantage of being an outside observer to the internal machinations of the Controller Hierarchy and is in a better position to monitor and diagnose the hierarchy's behavior, set goals, allocated resources, sequence activities, etc. These are same services the Controller Hierarchy provides to the real world.

Tasks that require a simultaneous, global look at the controller hierarchy can be difficult for the hierarchy to perform for itself, but are easily accomplished by a Meta-Controller. This is because each node in the hierarchy knows only of its parent (if there is one) and its children (if there are any). Tasks implemented or planned for the Meta-Controller include:

1. ***Prioritizing Simultaneous Planning Requirements:*** There will be times when numerous planners throughout the hierarchy are alerted by the monitoring and diagnostic function at the same time, or nearly

the same time. An arbiter is needed to prioritize the sequencing of these planning activities and initiate them.

2. ***Navigating the Hierarchy:*** The expansion of the planner hierarchy during the course of developing a plan can be performed in different ways. One is a recursive, depth-first traversal of the nodes in the tree, and can be managed easily by the hierarchy itself. But other navigation approaches, such as a breadth-first traversal or some hybrid approach, are not so easily performed from within the hierarchy—global view of the structure is required.

3. ***Command Triggering*:** The leaves in the controller hierarchy are those that generate the actual real-world commands. Commands are ready to be issued only when the plan from which they are derived is valid. The plan is valid when every node in the planner hierarchy is satisfied that it has solved the problem it was given. This may only occur after much give and take over many iterations between and among the levels in the hierarchy. So when is the plan complete and valid? And who knows about it? Again, this is determined easily by the Meta-Controller by monitoring the state of each controller in the hierarchy. When plans are valid, it can trigger the issue of associated commands.

4. ***Monitoring and Diagnosing Pathological Behavior*:** Pathological behavior in the dynamics of the hierarchy can be manifest in a number of different ways (lockup, thrashing, cycling, etc.), some of which could possibly be observed and treated within the hierarchy itself. But we believe that a systematic, comprehensive treatment of all the various bad behaviors that may come up are best handled by an agent outside the hierarchy, one that has a simultaneous, global view of what is happening at any given moment in time.

5.2.2.3 Internal Component Interfaces

One goal of the architecture is to simplify and standardize the interfaces between the various components in the hierarchy, so that individual components can be swapped in or out, depending on current requirements. For example, at times, a simple planner may be needed for its speed, while at other times, the accuracy of a more complex planner may be preferred. This ability to swap planners in and out was exercised in this project by using either the heuristic or optimal-based planner in performing experiments. It was also employed in our joint experiments with ALPHATECH, where simple planners were used to determine sensitivity information for negotiation, and then substituted with the detailed planners for the actual plan.

5.2.3 Concept of Operation

Much happens during the ongoing operation of a hierarchical controller. At each node in the hierarchy (although at perhaps at different times), state information is sensed, replanning decisions are made, portions of the controller hierarchy are rebuilt and reestablished, and commands are issued for execution in the real world (simulation). The replanning step itself is a multilevel orchestration of give and take as it converges to a solution satisfying the overall objective. An attempt at capturing the essential part of this stream of activity is shown in Figure 5.2-2. Here the action starts when a particular controller's monitor senses new state information, and ends when a new plan is issued and a new set of monitors are brought to life.

The events that are monitored at the different levels in the hierarchy and the criteria for branching in the diagram of Figure 5.2-2 are discussed below.

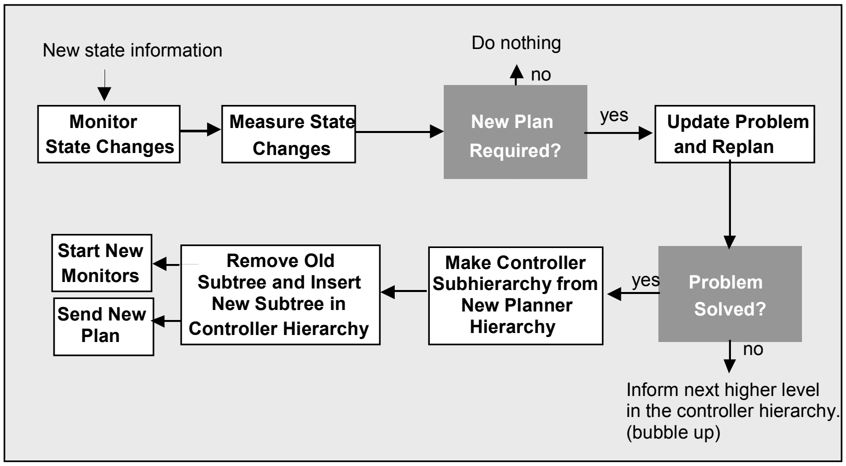

New state information
Do nothing
no
Monitor State Changes
Measure State Changes
New Plan Required?
yes
Update Problem and Replan
Problem Solved?
yes
no
Inform next higher level in the controller hierarchy. (bubble up)
Make Controller Subhierarchy from New Planner Hierarchy
Remove Old Subtree and Insert New Subtree in Controller Hierarchy
Start New Monitors
Send New Plan

Figure 5.2-2. Chain of Events Triggered by State Monitoring.

5.2.3.1 Monitoring Events, Diagnostic Criteria, and Bubble-up Criteria

During ongoing execution of the plan, all replanning activity in the controller is triggered by the state information read by the monitors and the judgement of their companion diagnosers. Monitoring events and diagnostic criteria currently in place are provided in Table 5.2-2. More can be added as additional modeling is incorporated in the simulation, and where in the hierarchy they should be addressed becomes known. Likely candidates for additional events include: weather changes, air base closures and openings, weapon shortages and surpluses, pilot unavailability, etc. Also included in the table are criteria for bouncing planning problems to higher levels when they cannot be adequately addressed locally.

Table 5.2-2. Monitoring Events, Diagnostic Criteria, and Bubble-up Criteria.

	Level 1 - Theater	Level 2 - Air base	Level 3 - Package
Monitoring Events	1. Number of new targets 2. Number of new time-sensitive targets	• Plan horizon expires • Loss of an aircraft • Package misses acquisition of its assigned target • Package fails to destroy its assigned target	• Assigned target moves substantially from its original position • The threat density map changes substantially
Diagnostic Criteria	• Replan directed if the number of above events exceed a threshold	• Replan required if horizon expires • Replan directed if the number of other events above exceed a threshold	• Replan directed if the change in either event exceeds a threshold.
Bubble up Criteria	• Not applicable	• The Planner failed to generate a new route for each aircraft in the package • New routes are substantially longer or shorter than the old routes • New routes are substantially more or less risky than the old routes	• None implemented

5.3 Controller Architecture Design Testbed

A hierarchical approach to system control makes the treatment of very complex systems tractable, but at the cost of more complexity in the control system. Orchestrating numerous subcontrollers to cooperate in doing the right thing requires careful design and extensive testing over the expected operating regime of the controller. To help address the issue of design complexity, we are implementing a controller development and test environment that offers the necessary assistance to the designer. This Controller Design Testbed will allow the designer to explore many more design alternatives quickly, offer better control over experiments and tests, provide enhanced diagnostic capabilities, and other services. A first version of the testbed, with limited but useful capabilities, is implemented and has been used to guide the design of our controller, and also in obtaining some of our experimental results.

Areas where the testbed can offer assistance include:

1. ***Broad and Deep Visual Feedback:*** Gain insight into behavior (normal and pathological) and performance with a range of models and scenarios. Display functioning of complete prototype controller. Go deep "behind the screen" to ferret out behavioral details.

2. ***Exploration of Design Alternatives:*** Facilitates experiments with different planners, monitors, diagnosers, and executors.

3. ***Control Over Experiments and Tests:*** Select scenarios designed to exercise planners and controllers in different ways. Choose problem size, configuration, world models.

4. ***Log and Display of Controller, Planner, and Simulation Performance Information:*** Interlevel transitions, reasons for replanning, timing (controller). Target value and cost versus time (planner). Number of sorties, attrition, retaskings, mission aborts (simulation).

5.3.1 Architecture

The basic structure of the testbed is shown in Figure 5.3-1 and indicates the main intent of the application. A powerful user interface is provided that allows the designer simultaneous views of activity in both the simulated world and the controller itself. It is possible to see the hierarchy being built, modified, and its internal status change as events occur in the simulation. This ability to visually connect what is happening in the "world" and what is happening in the controller goes far in helping to debug and develop robust controllers.

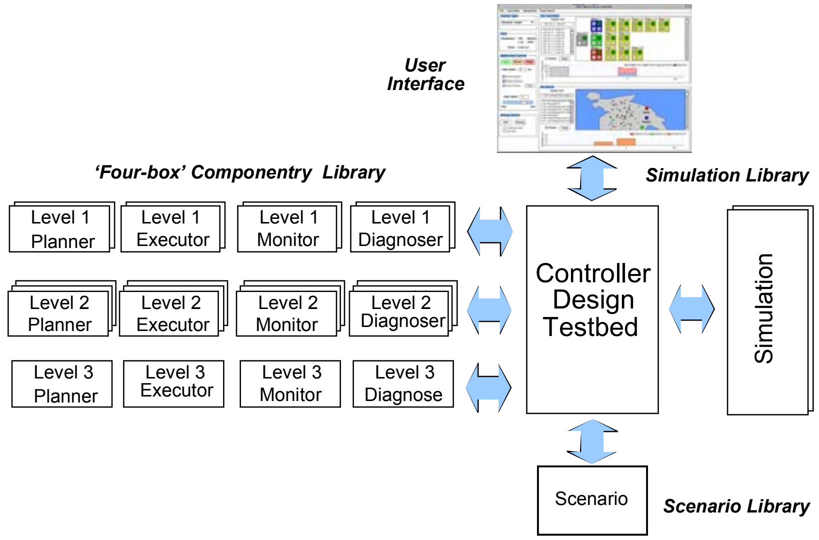


Figure 5.3-1. Controller Design Testbed Interfaces.

To facilitate a fluid and flexible design process, the testbed also shows interfaces to three important libraries of components:

1. ***"Four-box" Componentry:*** The library holds sets of Planners, Executors, Monitors, and Diagnosers that are compatible among themselves, and also compatible across different levels in a hierarchy. Arbitrary combinations of planners, monitors, etc., at various levels will not work well with one another.
2. ***Simulations**:* The ability to interface with simulations of varying scope and resolution is planned.
3. ***Scenarios:*** The ability to develop and access scenarios that exercise the controller in different is also planned.

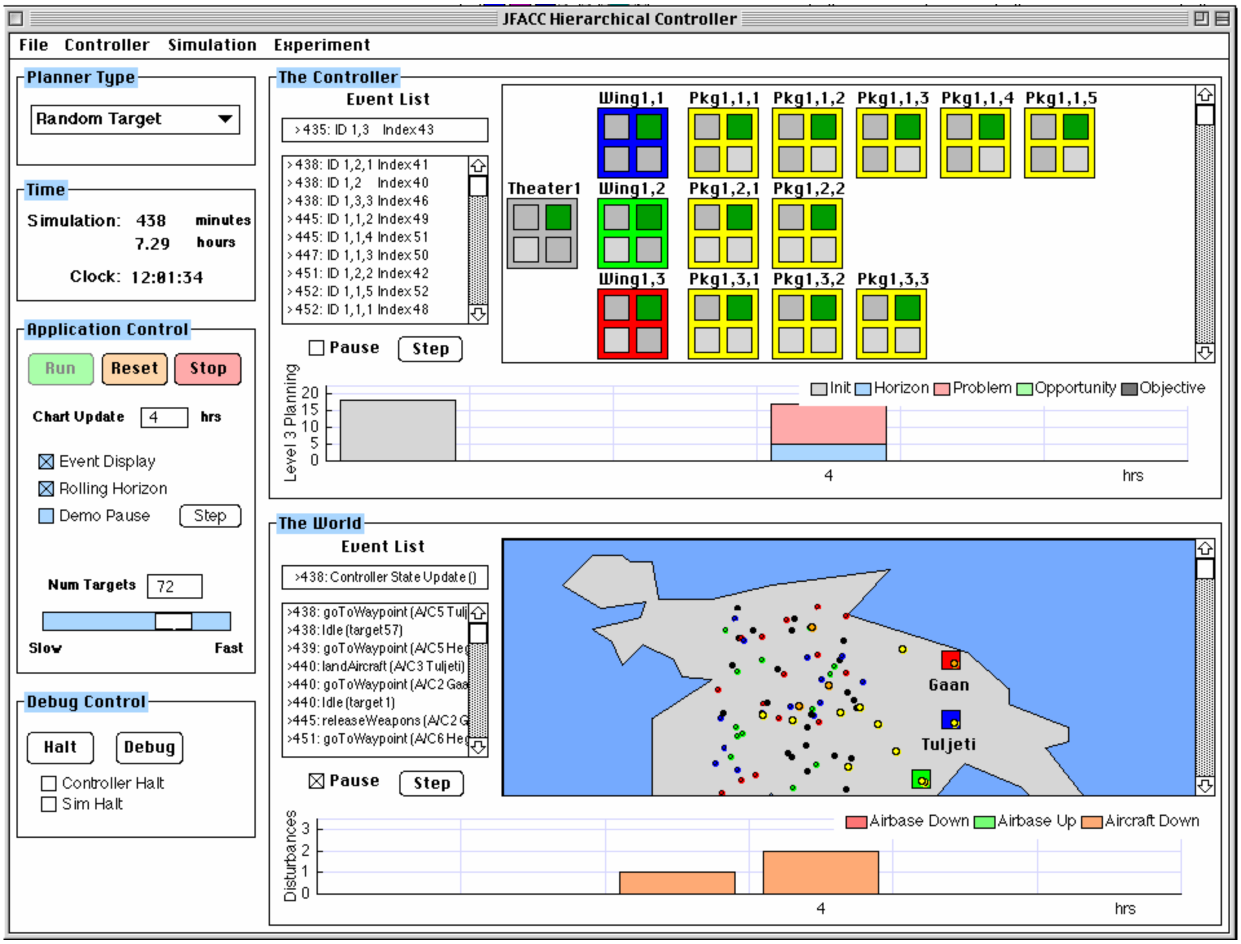
The easy control of specific architectures, scenarios, and simulations should help to dramatically increase the number design cycles that are important in developing good, robust designs.

5.3.2 User Interface

The current version of the application's main window is shown in Figure 5.3-2. The two main areas of the window are on the right and show basic information about the controller at the top and world at the bottom. In each case, a dynamic, graphical depiction of the important entities is shown, along with the event list for sequencing activities and a plot of pertinent activities over time. These two plots document the correspondence between those activities occurring in the world and those occurring in the controller. Also shown are buttons for pausing and stepping the controller and/or the simulation.

The vertical area on the left of the window provides the simulated and real time, and various controls for the application.

The application currently provides a number of other windows, one for setting up experiments, and others for displaying results. The results-window for the controller is shown in Figure 5.3-3, and shows planning activity plotted for each level in the hierarchy, along with statistics on planner timings, and interlevel transitions. Patterns in these kinds of charts can tell the designer how smoothly the controller is working, whether certain abnormal behaviors are beginning to creep in, the load balance among the levels, etc. As more is understood regarding the various pathologies that controllers can exhibit, additional pathology-specific displays can be added, as well.



Random Target

Simulation: 438 minutes
7.29 hours

Clock: 12:01:34

Chart Update 4 hrs

☒ Event Display
☒ Rolling Horizon
□ Demo Pause

Num Targets 72

Slow Fast

□ Controller Halt
□ Sim Halt

```
>435: ID 1,3   Index43
```

```
>438: ID 1,2,1 Index41
>438: ID 1,2   Index40
>438: ID 1,3,3 Index46
>445: ID 1,1,2 Index49
>445: ID 1,1,4 Index51
>447: ID 1,1,3 Index50
>451: ID 1,2,2 Index42
>452: ID 1,1,5 Index52
>452: ID 1,1,1 Index48
```

□ Pause

```
>438: Controller State Update ()
```

```
>438: goToWaypoint (A/C5 Tulj
>438: Idle (target57)
>439: goToWaypoint (A/C5 Heg
>440: landAircraft (A/C3 Tuljeti)
>440: goToWaypoint (A/C2 Gaa
>440: Idle (target1)
>445: releaseWeapons (A/C2 G
>451: goToWaypoint (A/C6 Heg
```

☒ Pause

Figure 5.3-2. Controller Design Testbed Main Window.

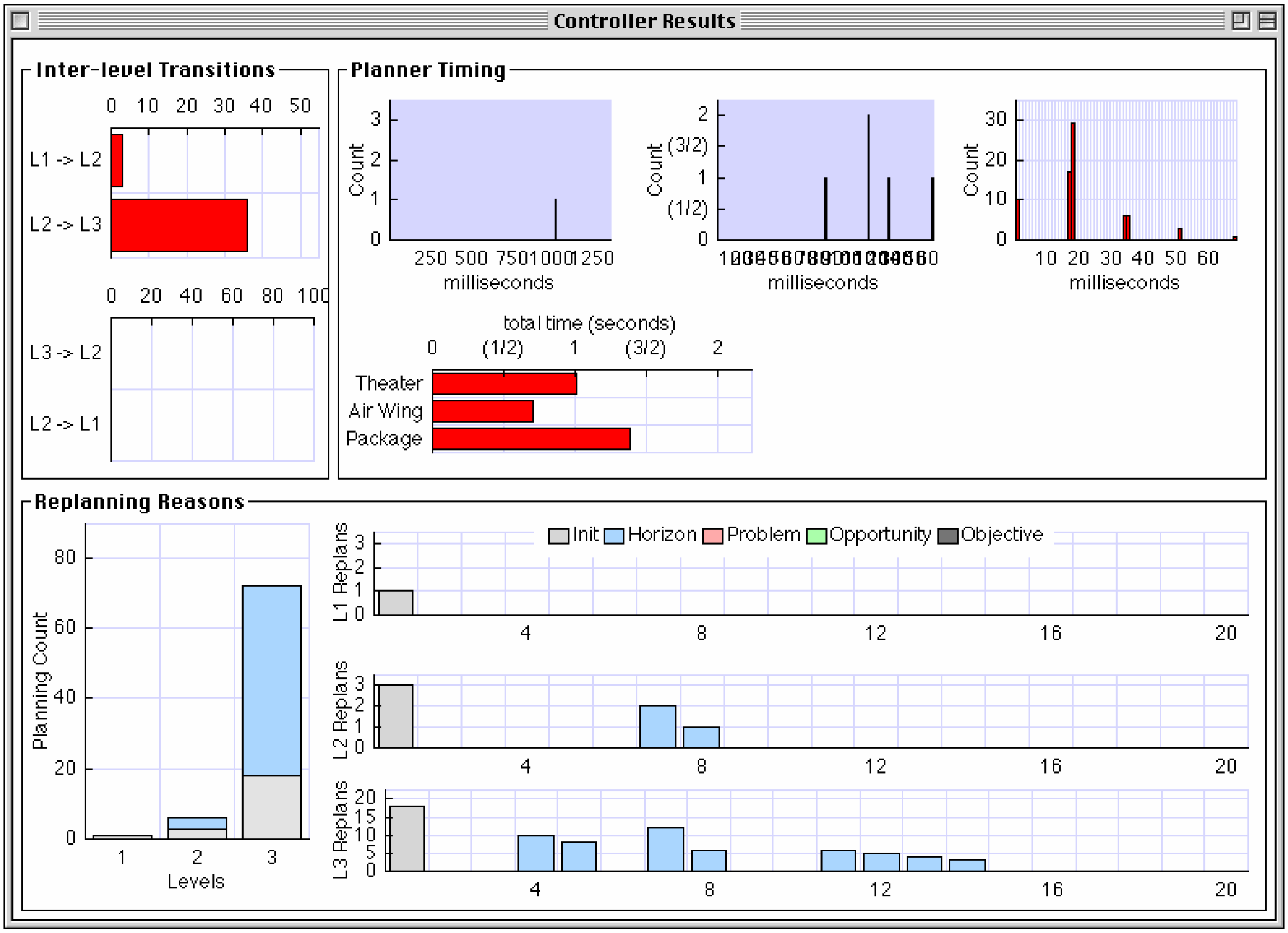


Figure 5.3-3. Controller Information and Statistics Window.

6 Modeling and Simulation Implementation

6.1 Controller-Plant Simulation Architecture

As illustrated in Figure 6.1-1, the JFACC Air Operations Controller and the associated plant can be viewed as a hierarchy of processes with some of these processes distributed over different spatial locations. This could be implemented as a hierarchy of simulated processes with associated controllers and the implementation could also be distributed. The current implementation, however, imposes the simplification of a single, global plant simulation that simulates the execution of the entire enterprise as a single computational process. Similarly, the current controller implementation, although inherently hierarchical in function, is implemented as a single computational process, separate however from the simulation process. The reduction in software complexity occasioned by this implementation decision greatly facilitates the execution of experiments as well as the initial development of all the simulation and controller software.

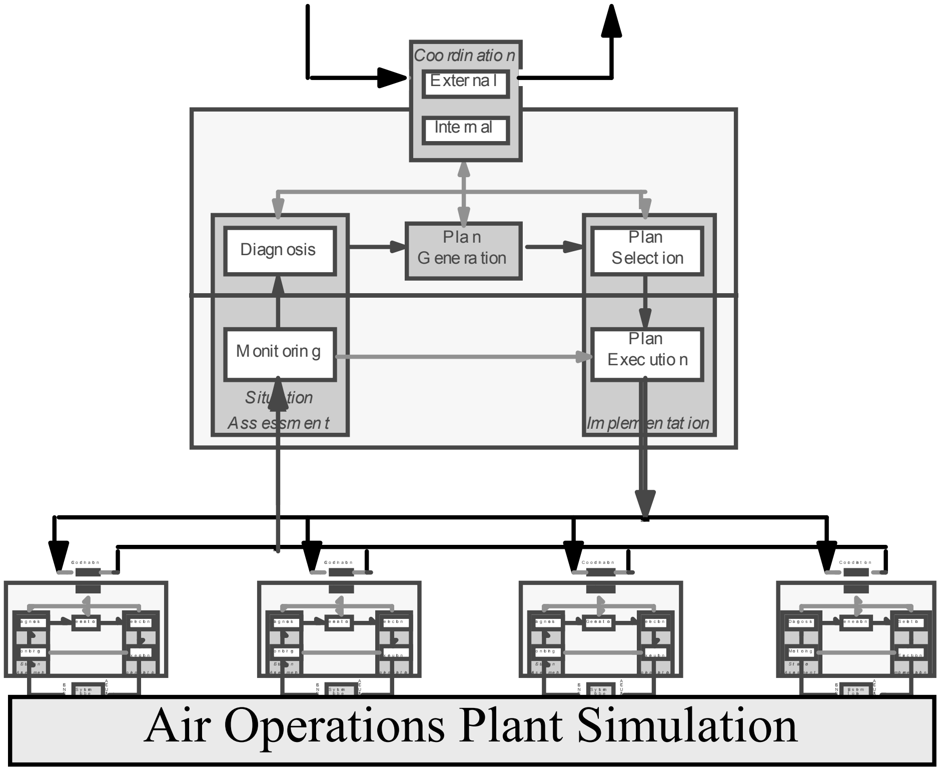


Figure 6.1-1. Hierarchical View: Aggregated Plant for Master Level

The state of the plant includes the following own force elements:

- Air bases.
- Tanker locations.
- Attack and strike package escort aircraft.
- Weapons.

The state of the plant relating to hostile forces includes:

- Targets.
- Air defense threats.

Finally, the environment is a spherical Earth geography that determines transit times given aircraft speeds and routes as specified by sets of waypoints. The air bases have a state that is either operational and available to launch missions or unavailable except for emergency recovery of aircraft. Refueling tankers are not explicitly modeled except by a location to which aircraft must fly to take on fuel. Tanker missions are not explicitly tasked, and tanker loadouts are not modeled. The tanker requirements that result from the Air Operations plan are recorded.

The state of strike aircraft includes the maintenance availability for mission tasking, the location, the fuel state, and the current weapons loadout. This is initialized to one of a set of standard conventional loads and then decremented on weapon release. Weapons are not modeled explicitly, but serve only to count capacitated resources.

Target state is modeled by a location, a current damage level, and a weaponeering specification that indicates how many of which weapon types are required to produce specified damage levels.

The air defense threat state is modeled only as a spatially-varying density of threat systems with associated engagement and lethality parameters.

The controller is assumed to have perfect information on own forces state, but may have incomplete or erroneous information on adversary targets and threats. Specifically, targets are disclosed at scripted times to the controller and may be available for attack only within a specified time frame. In the highly aggregated model, it is assumed that target acquisition opportunities vanish with a first-order exponential model from the time of target disclosure. Targets must be acquired before own force strike aircraft can release weapons and cause damage on those targets. Additionally, target weaponeering specifications sent to the controller may contain errors with respect to the simulated truth model. Finally, the estimated air defense threat density sent to the controller may contain discrepancies with respect to the simulated truth model. The effects of these discrepancies on controller performance are assessed in a specific set of experiments.

In summary, the classes of plant disturbances to be considered include:

- Unanticipated changes in own resources due to unexpected rates of attrition and unanticipated temporary base closures.
- Unanticipated level of effectiveness of own weapons employed.
- Unanticipated level of air defense threat effectiveness.
- Unanticipated changes in adversary activities reflected in changing target locations and values as reflected in the Commander's Intent.

Thus, disturbances are principally due to errors in modeling the evolution of plant state components; to uncertainty in exogenous inputs to the plant, such as adversary command and control directives; and to changes to own force command and control directives, such as changes in Commander's Intent weightings across phase boundaries.

6.2 Simulation

6.2.1 Simulation Functionality

The principal elements of our plant model are expressed by the primary aircraft activity functions:

- Ready Aircraft in a specified mission configuration and take-off on activity completion.
- Go To Location specified by a great circle route to the next waypoint, depleting remaining fuel endurance in transit.
- Refuel Aircraft if in proximity to a specified tanker location.
- Release Weapons specified for a particular target if in proximity to specified release location.
- Recover Aircraft if in proximity to an air base location.

Each of these activities has a logistical consequence, a specified activity duration, a specified earliest starting time and specified latest completion time, and a location proximity relation that must be satisfied to either initiate or complete the activity. Aircraft that arrive at a waypoint before the specified earliest start for proceeding to the next waypoint or activity execute a banked-turn loiter at that waypoint.

The air defense threat interaction model is evaluated for all aircraft in the air. One component of the model is the determination of physical proximity of defense suppression escort aircraft. A second component stochastically samples for attrition outcome from probabilities evaluated from the threat density, the threat parameters, the aircraft track, and the aircraft escort status. The details of this model will be discussed shortly.

The simulation implements controller commands, updating aircraft and target states according to the execution of planned activities. In the process of executing the missions, the simulation monitors fuel, pilot endurance, weapons, activity duration, proximity to base, tankers, and determines target damage status. Figure 6.2-1 is a snapshot of one of our simulation displays.

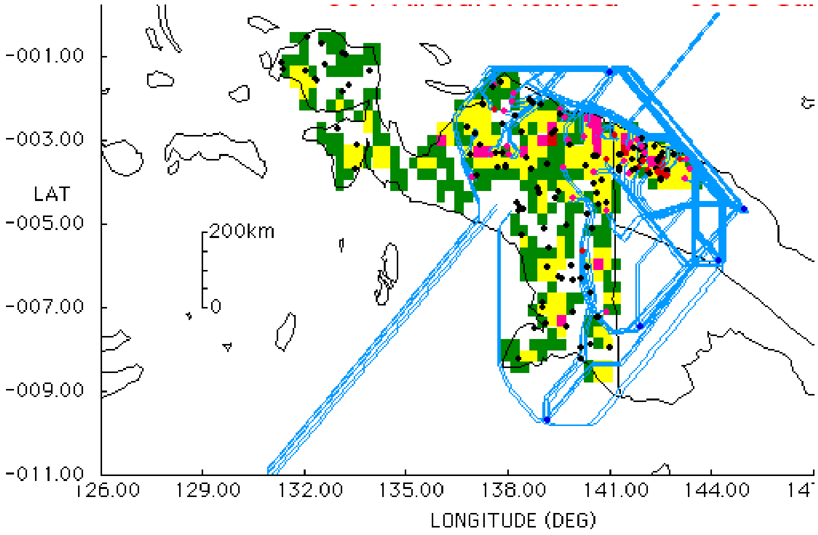


Figure 6.2-1. Simulation Display of Aircraft Routes.

6.2.2 Control Signals

The signals from our controller to the plant are contained in the command file, which consists of repetitions of one of five command types listed above for each aircraft that is to tasked or retasked. Depending on the specified planning horizon, the command file may contain several sets of tasks for particular aircraft representing successive sorties.

An example ReadyAircraft command is:

```
[AircraftPlan]
aircraftIDType          HAWK-3-5
activityType            ReadyAircraft
activityID              328
earliestStart           0
latestEnd               2700
minDuration             2700
mission_configuration   F-5-010
role                    striker
```

The mission configuration indicates the standard conventional load. The definition of F-5-010, for example, is contained in the state file sent to the controller:

```
[AircraftType]
mission_configuration               F-5-010
generic_type                        F-5
role                                striker
[endlist]
maxFuelEndurance                    8823.5
aveSpeed                            550.0
maxPilotEndurance                   26470.6
nominal_turn_time                   7200.0
ground_prep_time                    2700.0
midair_refuel_time                  400.0
time_to_major_maintainence          360000.0
maintainance_time                   120000.0
munition                            MK-2
number                              2
release_duration                    180
[endlist]
```

The ultimate objective for this mission is revealed after a sequence of waypoint commands, culminating in a Release Weapons command on target BE number 1515:

```
activityType                        GoToLocation
activityID                          336
earliestStart                       4627.48
latestEnd                           4672.44
minDuration                         44.9671
latitude                            -5.2053
longitude                           138.645
altitude                            12000

activityType                        ReleaseWeapons
activityID                          337
earliestStart                       4672.44
latestEnd                           4852.44
minDuration                         180
beNumber                            1515
munitionType                        MK-2
number                              1
[endlist]
```

Additional waypoint commands would culminate in Refuel Aircraft activities and in the Recover Aircraft activity at an air base. Examples of these commands are:

```
activityType                        RefuelAircraft
activityID                          1005
earliestStart                       24800.7
latestEnd                           25200.7
minDuration                         400
```

TankerID	SHELL
activityType	RecoverAircraft
activityID	1013
earliestStart	28375.6
latestEnd	28375.6
minDuration	0

This is the entirety of control signals sent from the JFACC Air Operations Controller to the JFACC Air Operations Simulation.

6.2.3 Aircraft Attrition from Air Defense Threats

The primary dynamic of Air Operations is bringing weapons to bear on known targets by tasking and scheduling flight operations over considerable distances and time durations. This dynamic is limited by the availability of aircraft resources and by the desire to prosecute offensive operations while limiting losses to enemy air defenses. Effective air operations cannot be sustained in the face of *high attrition* losses, where *high attrition* was considered to be several percent per raid in the context of 300-bomber raids of World War II and is currently an order of magnitude lower.

Tactics for mitigating attrition to air defenses, in addition to defense suppression, include active targeting of air defense elements, especially C^3 elements, during the early phase of a campaign. Sustained operations focused on these elements will roll back the barriers to accessing targets that were initially highly defended. These tactics can be implemented in the current JFACC Air Operations Controller by employing the Commander's Intent weightings to emphasize air defense target categories during the early part of the campaign.

It may not be reasonable to demand that operations be mounted without any losses, especially since the targets most valuable to both sides are likely to be most heavily defended. What is reasonable is the use of attrition models in operations planning, adapted to a posteriori experience, to include the effects of enemy air defenses on the likelihood of incurring attrition during the execution of missions. Attrition models are used by the controller in two ways:

1. Assessment of relative risk between alternative selections of targets, packages, and routes to pick the lowest risk selection.
2. Estimation of absolute risk to be used to deselect or defer missions otherwise selected for the purposes of risk management.

In *neither* case is it necessary for the attrition model to present a high-fidelity rendition of "reality," if that were in fact possible. For the former application, the controller attrition model needs only to contain enough domain acuity to be able to order the relative risk between alternative plan selections to support systematic search of different escort packaging and different routing to different targets. Essentially, extended traversal through high-threat density areas needs to be avoided at the planning level. For the risk management objective, the omissions and uncertainties in the absolute attrition estimates produced by the model can be mitigated by treating the risk management threshold as a free parameter. Uncertainties with respect to ground truth would require this approach, even if high-fidelity modeling were employed.

The actual domain in which air defenses cause attrition is rife with complexity, including hierarchical organization with sector and subsector commander logic, networked surveillance radars, as well as fire control radars, firing doctrine including salvo size, and large numbers of man-portable surface-to-air missiles (SAMs) and antiaircraft artillery independently engaging aircraft flying low or engaging in low-altitude delivery tactics. From the perspective of Air Operations planning, it is necessary to be cognizant of those aspects of the overall threat environment that are knowable, but at the same time, it is fruitless to dwell on minute details of terminal area delivery tactics for the thousands of missions that are to be planned and coordinated. The decentralized execution concept correctly removes that responsibility from Air Operations planning. It would be similarly fruitless to incorporate highly detailed threat models in the JFACC Air Operations Controller, since most of the parameters are unknowable and the resulting impact on calculation time would be untenable.

The effect of *estimated* attrition risk on Air Operations planning is to cause the following:

- Avoidance of possibly lucrative targets that are judged too risky to prosecute.
- Selection of threat avoiding routing that increases mission timelines and use of tanker resources and otherwise reduces operational tempo.
- Reduced ability to quickly transit to time-critical targets.
- Increased demand on escort resources and reduced target prosecution rate from escort constraints.
- Increased use of scarce and expensive stand-off weapons.

The effect of actual aircraft attrition is to reduce the resources available to prosecute targets, and especially to reduce possibly scarce and limited escort resources such as jammers. In addition, plan adjustments are necessary for both in-progress missions where aircraft have been lost, as well as for downstream missions that were planned using those aircraft. The JFACC Air Operations Simulation needs to communicate threat information to the controller and also to simulate attrition in the execution of missions commanded by the controller. If the controller is to be able to systematically take risk from air defense threats into account in the planning process, there needs to be some correlation between attrition models employed by the simulation and those employed by the controller. Identical models would illustrate the upper bound of achievable performance with respect to risk. Discrepancies between simulation and controller models represent the case where the simulation is a surrogate for reality and provides an assessment of how well the controller responds to the stress of modeling errors that lead to higher than anticipated losses as well as the opportunities created by lower than anticipated losses.

The underlying basis for the aircraft attrition model in the JFACC Air Operations Simulation and the JFACC Air Operations Controller is similar in that air defense threats are represented by a spatially varying probability density distribution. The relation of the density of threats to the likelihood of engagement, and further, to the likelihood of aircraft loss is also similar. The simulation, however, samples stochastically for the binary outcome of this set of processes, whereas the controller uses the analytic estimate of probability of attrition as one of the primary planning inputs. The controller uses the analytic estimate to form a cost function that is used for routing and it uses the path-integrated attrition probability to select from alternative candidate mission plans.

The underlying models that relate the threat laydown and assumed parameters to probability of attrition are as follows. The attrition caused by SAMs is a function of:

- The density of SAMs over which a mission is routed.
- The state of the air defense C^3 system that provides cueing for these systems.
- The presence of defense suppression including escort jammers and HARM-equipped Wild Weasels.

Defining the following quantities,

ρ^i = density of air defense systems of type i (systems/km^2)

R^i = effective engagement radius (ground range) (km)

NPE^i = number of potential engagements (NPE) by systems of type i

$P^i_{E|E}$ = probability of engagement given engageability

$P^i_{K|E}$ = probability of kill given engagement the density would be specified to a granularity of 30 km per grid box as illustrated by the color coding in Figure 6.2-1 above and Figure 4.7.2 in Section 4.7. The NPE traversing a flight path over a random distribution of these threats with specified density in each area is given by

$$\mathrm{NPE}^i = \int_{\mathrm{path}} dr \left[\rho^i \;\; 2R^i\right]$$

The decision to engage is modulated by the presence of wild weasels. Some SAM operators may decide to be heroes, but many, perhaps one half to two thirds, will withhold fire after observing the smoking ruins of the adjacent SAM operator encountering fire from a HARM-shooter (aka "wild weasel"). Hence, the probability of engagement given engageability multiplies NPE to determine the number of engagements along a path traversal. The overall probability of attrition, given that an aircraft can only be killed once, is given by:

$$P_K = 1 - \sum_i \left[1 - P^i_{K|E}\right]^{\mathrm{NPE}^1 \, P^1_{E|E}}$$

This formulation does not assume detailed knowledge of the location of individual threats, but employs a statistical model for the likelihood of being engaged by any one of a number of threats where a threat is assumed to have an effective engagement radius that may vary with threat type. The model estimates the likelihood of passing close enough to threats to trigger an engagement. The outcome of an actual engagement is specified by the probability of kill given an engagement, $P^i_{K|E}$. The numeric value of $P^i_{K|E}$. includes credit for the intrinsic effectiveness of the defensive system type i against aircraft with chaff, flares, towed-decoys, and other self-defense systems. In other words, the P_k may be closer to 0.01 in operational use than to 0.7 against drones in level flight.

The effects of jammers and local destruction of Integrated Air Defense (IAD) C^3 is reflected in the model for R^i, where the kinematic ability of the SAM to intercept a target flying rapidly through its engagement zone is reduced by acquisition delays that can be caused by destruction of cueing from long-range search radars and by jamming. The effective engagement radius can shrink to a minimum value that is given by optical sighting. An example functional form that produces the correct qualitative features is displayed in Figure 6.2-2.

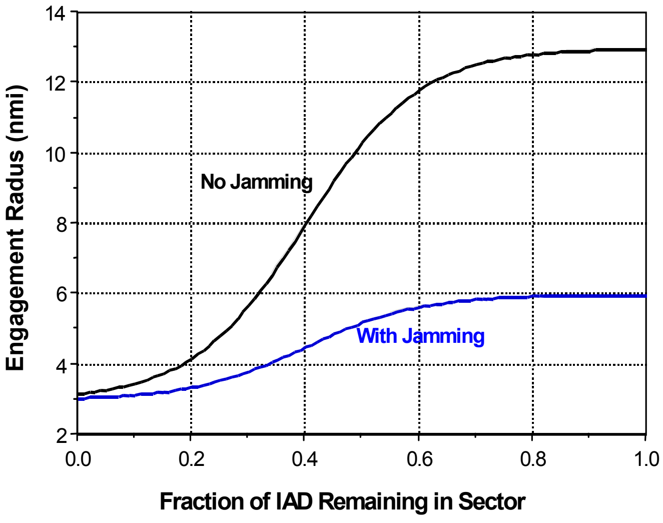


Figure 6.2-2. Engagement Radius Variation with IAD Remaining and Jamming.

In most cases, the greatest operational effect of escorts is defense suppression as outlined above. The reduction in threat density resulting from destroying SAM units is not currently modeled in the JFACC Air Operations Simulation since this is expected to be a higher-order effect without real importance to the development and experimentation with the JFACC Air Operations Controller. The marginality of this effect was confirmed by the Air Operations Enterprise Model (AOEM) calibration experiments, where attrition of air defenses was modeled explicitly. Avoiding this model avoids defining additional, difficult to validate complexity.

6.3 Draper JFACC Cyberland Scenario

This section will provide more detail to the scenario overview presented in Section 4.7. Those elements of the scenario that are most important are:

- The deployment of a total of 80 strike and escort aircraft to five bases.
- The definition of three aerial refueling stations.
- The definition of a nominal case of 313 targets and an excursion of 910 targets with gradual disclosure of targets over 5 days of operations.
- The definition of a distribution of air defense threats.
- The definition of details, including target weaponeering, aircraft standard conventional loads, Commander's Intent specifications, etc.

6.3.1 Scenario Context

The original scenario storyline included an invasion by West Cyberland, a former Soviet-bloc client with a large military and inventory of weapons, of East Cyberland, a U.S. ally with a modest military and significant production of oil in the North central portion of the country. The invasion forces meet with little resistant and quickly establish an invasion salient headed toward the East Cyberland oil fields.

The top-level U.S. objectives are to halt and roll back the invasion, preventing seizure of or damage to East Cyberland oil fields, if possible. The U.S. has a couple of aircraft carrier battle groups (CVBGs) at distances requiring 2 and 5 days steaming to the theater of operations. It is assumed that tactical strike aircraft can be staged into several East Cyberland in-country bases that are not close to being overrun to support high-tempo operations against the invasion and against supporting infrastructure in West Cyberland. Additionally, bases in Guam and Darwin, Australia, at about 2 h flying time from the theater, are assumed available for staging tanker, long-range bomber, ISR, and other supporting missions.

6.3.2 Deployed Resources

At the time that air operations commence on Day 0, we assume that the invasion salient is halfway to the oil fields, and that the better part of an Air Expeditionary Wing has been deployed as shown in Table 6.3-1.

Table 6.3-1. Deployment of Aircraft to Bases.

Base	Gaan	Mt. Hegan	Tuljeti	Darwin	Andersen AFB Guam
ACType					
F2W	14	8	4	0	0
J2	5	2	2	0	0
F5E	18	12	6	0	0
B101	0	0	0	3	0
B102	0	0	0	0	3
B100	0	0	0	0	3

Aircraft from the aircraft carriers are not played in the scenario. The most limiting resource is jammers, the type "J2," with the 9 deployed split between the major base at Gaan and the two secondary in-country bases. The aircraft type F2W is the HARM-shooter or wild weasel used alone or with jammers as a defense suppression escort to F5E strike aircraft. There are 3 of each of 3 types of bombers, with two groups deployed to Guam and one to Darwin. Counter air escorts, tankers, airborne warning and control system (AWACS), and transports are outside project scope and are not explicitly played in the simulation. In actuality, tanker tracks would need to run North-South about 100 to 150 km from the West Cyberland/East Cyberland border and the combat air patrol (CAP) would need to be spread along the border and also to cover tanker, AWACS, and heavy bomber missions out of Guam and Darwin.

Initial experimentation used a total of 51 deployed aircraft. The number was increased to 80 aircraft, in consultation with Program Office air operations subject matter experts, in order to provide more assets for a robust operational tempo that would address most of the 313 targets within several days of campaign time.

The ability to prosecute targets depends not only on the number of aircraft, but also on the number of weapons required to achieve specified damage levels on targets (i.e., the weaponeering specification) and also on the carrying capacity of those aircraft as reflected in the set of standard conventional loadouts (SCLs). The F5E generic aircraft type was assumed to be capable of being configured in one of five different SCLs (where CBUxx and GBUxx are types of munitions):

- F5E03 with 8 CBU87s.
- F5E06 with 4 GBU10s.
- F5E07 with 4 GBU16s.
- F5E10 with 4 GBU31s.
- F5E13 with 2 GBU16s and 2 GBU32s.

The bomber SCLs were limited to no more than 8 munitions of any particular type because the algorithms for scheduling multiple target tours for heavy bombers were not integrated into the controller.

The target weaponeering was simplified to require between 1 and 5 weapons of either the type CBU87 or type GBU31 to produce the specified target damage effect for different types of targets, with the weaponeering error excursion sending erroneous information to the controller that only 1 to 3 weapons were required. The initial specifications for weaponeering were more elaborate, but they were reduced to just 2 weapon types to simplify the interpretation of experimental controller behavior.

The set of possible strike packages included one or two strikers in various SCLs, the same with weasel-only F2W escorts, and packages with strikers, weasels, and jammers. Since there were no defense suppression escorts deployed to Guam or Darwin, bomber packages included 2 heavy bombers of any given type. Assembly points were not generally used, although some excursions with assembly points enabled bombers to form up with escorts before entering defended airspace.

The AOEM scenario used with the JFACC Air Operations Controller applied a weaponeering requiring many more weapons per target and also used a larger number of packages, including cases with many more strikers.

6.3.3 Air Defense Threats

The original Cyberland scenario had 16 bases and locations in West Cyberland and 11 bases and forward operating locations in East Cyberland. It was specified that there were 14 EW/GCI sites, 24 fixed SAM sites, 9 Sector Operations Center/Air Defense Operations Center (SOC/ADOC) sites, 330 air defense artillery sites, and 260 mobile SAM units. The mobile SAM units (assumed SA-6 with one self-propelled radar vehicle and 4 self-propelled launchers with 3 missiles each) were assumed to be allocated principally to Army units. The Army units were specified as a total of 3 Corps, with 1st and 2nd Corps constituting the invasion salient and 3rd Corps in reserve in the Port Manley area. A guess at a representative allocation of SAM systems is:

- (80) systems in 1st Corps (mostly armor).

- (50) systems in 2nd Corps (mostly infantry).
- (80) systems in 3rd Corps.
- (18) systems at miscellaneous locations.
- (32) systems with 2 at each of 16 key locations.

Hence, there would be 130 mobile SAM systems moving with the invasion salient whose spatial locations were not precisely known and whose distribution would be reflecting the area covered by the invasion force.

Threat systems were assumed clustered around airfields and the remainder of the 260 systems indicated in the original scenario were sampled from a geographic distribution within polygons defining the Cyberland scenario regions (e.g., Western Air Defense District, Northern ADD, Southern ADD, and invasion salient) with 40% of all threats sampled within the Northern ADD, 20% of the 260 threats within the Southern ADD, 30% in the invasion salient, and 10% in the Western ADD.

Figure 4.7-2 in Section 4.7 illustrates the discrete threat locations and the corresponding density distribution representation. The location of the threat systems was sampled in the vicinity of locations that were identified as targets on the theory that areas considered targets by us are considered worthy of having defenses mounted. As can be seen from Figure 4.7-2, some areas were pretty well surrounded by high density of threat systems, whereas targets in other areas were more accessible without incurring attrition risks. In general, it is necessary to pass through some regions containing threats on the way to targets, with the objective of the router being to minimize this exposure. The scenario is intended to capture the essence of this problem.

The threat model parameters for the nominal scenario are given by the following values:

```
[Threats]
Threat_type                   SAM1
Engagement_Radius_Max         30.e3
Engagement_Radius_Max_Jam     5.e3
Engagement_Radius_Min         2.e3
Pe_No_WW                      0.50e0
Pe_WW                         0.25e0
Pk_Given_Engagement           0.01
```

The Pk_Given_Engagement was chosen empirically to yield what was considered to be a realistic attrition rate of a fraction of a percent per sortie and leading to the loss of up to a handful of aircraft over a weeks worth of air campaign, including 1000 to 2000 sorties.

6.3.4 Target Distributions

Target locations, with the exception of the fixed airfields, were sampled from the same geographic distribution as that list above for the air defense threats. Additional sampling was employed to:

- Distribute targets between functional categories.
- Distribute target disclosure over time.
- Distribute the weaponeering requirements.

The target distribution over time was:

Day	Number of Targets	Percent of Total
1	156	50
2	32	10
3	63	20
4	32	10
5	15	5
6	15	5

indicating about half of the targets to be known at the start of operations with the remainder disclosed over the next five days.

The target distribution across geographic regions was:

Region	Percent of Total
Western AD District	10
Northern AD District	40
Southern AD District	20
WC Invasion Salinet	30
Commander's Special	0

The geographic distribution of targets is illustrated in Figure 6.3-1.

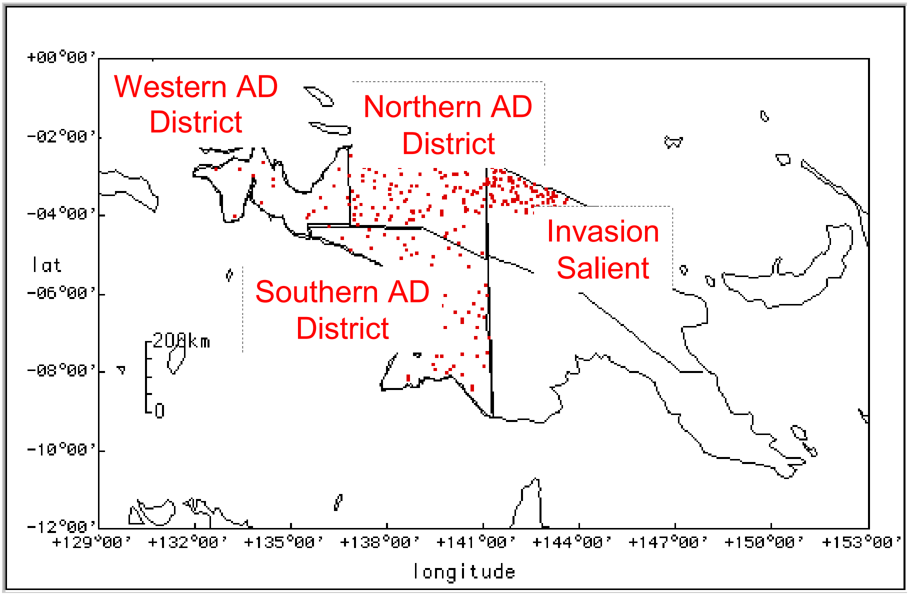


Figure 6.3-1. Distribution of Targets.

The distribution across functional categories was conditioned by geographic region, as shown in Table 6.3-2. The fractional entries in each column are normalized to unity and multiplied by the number of targets in those regions to derive the number of targets in a particular "cell" of the table.

Table 6.3-2. Distribution of Targets Across Functional Categories and Geographic Regions.

Target Functional Category	West_ADD	North_ADD	South_ADD	WC_salient	Commander's Special
IAD_C3	0.26	0.15	0.25	0.20	0.00
IAD_nonC3	0.10	0.15	0.25	0.10	0.00
Other_C3	0.20	0.25	0.25	0.20	0.00
Airfields	0.10	0.13	0.00	0.10	0.00
LOC	0.00	0.10	0.00	0.10	0.00
POL	0.13	0.03	0.05	0.00	0.00
Man_units	0.00	0.10	0.00	0.10	0.00
LRarty	0.20	0.10	0.20	0.20	0.00

The time-critical targets were defined as all of those targets in the cells indicated in Table 6.3-3.

Table 6.3-3. Time Constants for Time-Critical Targets.

Target Functional Category	West_ADD	North_ADD	South_ADD	WC_salient	Commanders Special
IAD_C3					
IAD_nonC3				14400.	
Other_C3					
Airfields		43200.			
LOC					
POL					
Man_units				14400.	
LRarty				14400.	

In other words, targets in the Man_units category in the WC_salient were time-critical, and the time constant for the first-order model described in Section 4.7 was 14400, or 4 h. The total fraction of time-critical targets was about 0.17, with experimental excursions defining all targets in all cells as time-critical targets and also eliminating all time criticality of targets.

The assumed distributions for targets across time, geography, and functional category were constructed to stimulate the JFACC Air Operations Controller with a diverse but structured distribution of targets across the battlespace. The ability of the Commander's Intent inputs to focus or direct the controller to particular elements of the battlespace was assessed in a number of experiments.

6.3.5 Target Effects Modeling

Target effects, as described in Section 4.1.3, were implemented as deterministic processes. If the requisite weaponeering specification of numbers of weapons were released on a target, the specified damage level was assumed to be achieved. This is consistent with a conservative weaponeering that attempts to minimize the number of missions that must revisit partially damaged targets in order to accomplish the damage objectives. It also served to avoid the introduction of statistical variability into the experiments and left the attrition process as the only stochastic element of the simulation.

6.3.6 Monte Carlo Samples

The JFACC Air Operations Simulation provides for the use of up to 1000 statistically independent pseudorandom number strings. Different processes in different objects can be assigned their own distinct pseudorandom number strings so that the addition of new objects or processes does not perturb the sampling for existing objects and processes. This capability is known as correlated sampling and is a powerful tool for reducing the variance in perturbation studies.

Although threat locations and target characteristics were sampled from distributions, this was performed as an off-line process, and results were stored to file for identical application in each of the experiments to be described. As mentioned above, the only stochastic process in the current simulation is the sampling for the possibility of air defense threat engagements and their outcomes. Because the closed-loop controller reacts to attrition events by retasking and rescheduling missions, the future evolution of an experiment is uniquely altered once an attrition event has been sampled. The results of five different Monte Carlo samples for the baseline experiment is shown in Figure 6.3-2. The computed variances amounted to several percent. An empirical analysis reveals that the number of aircraft attrited appears to follow a Poisson distribution with the standard deviation being equal to the square root of the mean number attrited.

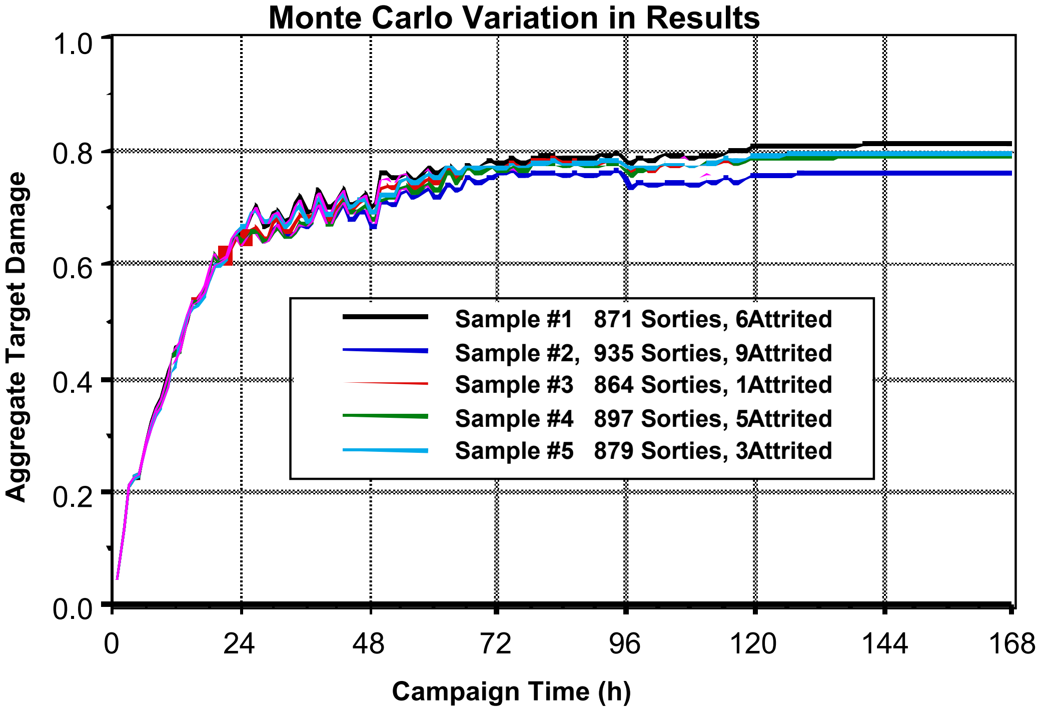


Figure 6.3-2. Monte Carlo Results for Baseline Experiment.

7 System Integration - Controller and Simulation Interfaces

As shown in Figure 7-1, the JFACC Air Operations Simulation and the JFACC Air Operations Controller interface to each other through a set of shared interface files, and there are also several other files that are integral to the definition and interpretation of experiments. The sequence of operation is as follows:

- The user prepares the Initial Data or Scenario file for reading once by the simulation.
- The simulation generates a state file to be sent to the controller.
- The controller generates a command file that is read by the simulation.
- The simulation executes the command file, generates a number of output files for analysis, debugging and replay, and repeats the cycle of sending the state file to the controller.

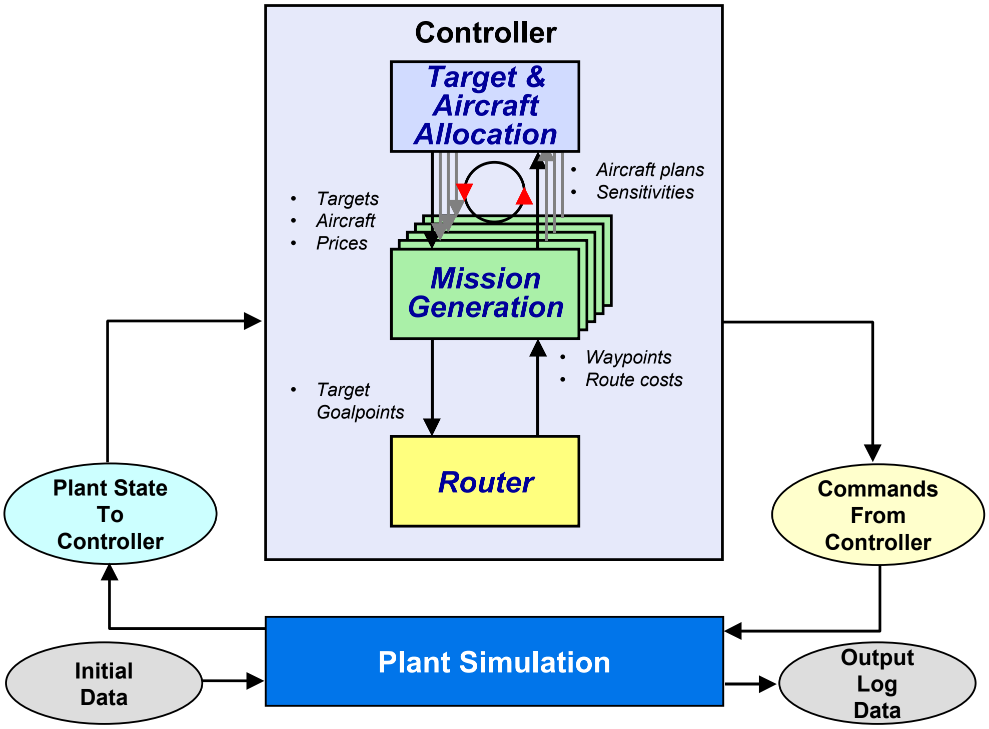


Figure 7-1. Controller –Simulation Architecture.

There are additional files required by the simulation that do not change from experiment to experiment including:

- World vector shoreline and political boundary files for map visualization.
- File of random number seed values.
- File of waypoints and specifications for externally scripted objects.

The externally scripted objects file is not changed between experiments since this capability is not currently used in the experiments that we have defined.

Everything needed to define a scenario is included in the scenario file. The data items will be described in detail shortly, and will be seen to be very similar although not identical to the data items on the state file.

Finally, to facilitate the unattended execution of a number of long-running experiments, a "Production Run" file is defined that specifies the filenames of a number of scenarios along with perturbation and Monte Carlo parameters.

7.1 Controller Memory

The essential architectural feature of the Draper interface design is that *everything* that the controller needs to make decisions and issue commands is contained in the "state" data sent from the simulation. In other words, the "state" file that is issued to the controller contains every piece of information that can be used by the controller *at each instance of its issuance* to the controller by the simulation. If the purely cyclic version of the controller is executed, there is no controller internal memory relating to state variables, and a complete and entire set of commands is issued for all aircraft on every instance of the command file. If the event-based controller architecture is executed, then the controller may issue commands for only the subset of aircraft whose plans are to be changed, and the simulation continues execution of the previously issued commands for aircraft not referenced in the current command file.

To maintain continuity of mission plans for en route aircraft across instances of controller invocations, the current target objective for each aircraft mission is encoded in the state file and is used by the controller, along with the "retasking" cost, to decide whether to continue with the current mission objective or to pursue other objectives, including a possible return to base. A high retasking cost will discourage changes, whereas a low cost will enable en route aircraft to be retasked to address newly emerged time-critical targets or fill in gaps left by attrition.

7.2 Activity Duration

Recalling the grammar for controller commands from Section 6.2.1, that is, the definition of five high-level activities including: ReadyAircraft, GoToLocation, RefuelAircraft, ReleaseWeapons, and RecoverAircraft, the finite duration of commanded activities requires that some provision be made in the interface design for maintaining continuity of in-process activities. Since every commanded activity has a duration, and activities have arbitrary start and end times, each cycle where the simulation dispatches the state data and waits for a controller plan will be occasioned by a number of activities that are “in process” at that particular time. For example, an aircraft may be 90% into its planned duration for a refueling activity. To properly account for the state change on completion of the activity with a minimum of mechanization complexity, the following approach is defined. The subset of activities that are of short duration and would normally be completed regardless of new longer-range plans are defined to be "uninterruptible." These include refueling, releasing weapons at the weapon release location, and landing. At the time when the simulation sends the state to the controller, the aircraft engaged in these uninterruptible activities have their states advanced to reflect the completion of the current activity, and they also have a variable called "earliest availability" set to the time of completion. This variable is used to indicate the time when any object will become available. Examples are:

- When aircraft are staged into theater over time.
- When aircraft come out of maintenance.
- When tankers come on station.
- When targets are disclosed over time.

In this manner, the controller can properly construct plans, including aircraft that are in the middle of short-duration activities when the controller is planning.

7.3 Description of State Information

A series of tags are defined to provide a "lite" version of extensible markup language (XML) functionality, where square brackets to delimit the tags. A parser searches for recognizable tags from a list that is hard-coded. The format after the tag is two space-delimited fields on each line, the first being a human-readable character string descriptor and the second being the data value. Blank lines are skipped. The number of data lines and the particular information that is expected after each recognized tag depends on the type of the tag and is hardwired into the tag reader procedure. If a tag type is in the file but is not recognized, the contents associated with that tag are skipped. The data descriptors are quite helpful in the debugging of simulation and controller issues.

The list of state file tags actually used includes:

[RunIdentifier]
[AircraftType]
[Aircraft]
[PackageConfiguration]
[WeaponType]
[TankerOrbit]
[Airbase]
[Target]
[CampaignPhases]
[Commanders_Priority_Table]
[Valuation_Type_Table]
[Pd_Threshold_Table]
[Perishability_Time_Table]
[Reconsitution_Time_Table]
[CostMap]
[Threats]
[Table]
[MaximumMissionRisk]
[LogCostPerFlightHour]
[UnitFlyAwayCost]

The **[RunIdentifier]** tag contains information on the simulation version, run date and time, simulation time, experiment descriptor, and Monte Carlo trial number.The **[AircraftType]** tag is repeated for each aircraft type and includes the following items:

```
[AircraftType]
mission_configuration          F2W00
generic_type                   F2W
role                           weasel
[endlist]
maxCombatRadius                400.
aveSpeed                       450.
maxPilotEndurance              28800
nominal_turn_time              7200.
ground_prep_time               2700.
midair_refuel_time             400.
time_to_major_maintainence     360000.
maintainance_time              120000.
munition                       AGM88
number                         4
release_duration               180.
[endlist]
```

Multiple mission configurations are typically defined for each generic aircraft type, the configuration specifying the weapon load ("Standard Conventional Load") and the possibly load-dependent combat radius in nautical miles and average speed in knots. The times are all listed in units of seconds and include all of the relevant activity durations and logistical times. The [endlist] tags indicate termination of n-tuples of information. In the example above, the first [endlist] terminates the 1-tuple list of "roles," and the second [endlist] terminates the list of 3-tuples indicating munition name, number carried in that configuration, and release duration in that configuration.

The **[Aircraft]** tag is repeated for each aircraft and includes the following information:

```
[Aircraft]
tail_number                   TIGER-2-0
mission_configuration         standby
generic_type                  F-2E
airbase                       Mt_Hegan
FuelEndurance_remaining       0.
PilotEndurance_remaining      0.
currentLatitude               -5.87360
currentLongitude              144.23330
currentAltitude               200.
assignedTargetBE_Num          0
beginAvailability             0.
endAvailability               10000000.
time_activity_begun           -1000000.
time_remaining                1000000.
missionReadyState             1
munition                      none
number                        0
[endlist]
```

The [endlist] tag refers to the munition type and number 2-tuple list. The missionReadyState field indicates 0 for aircraft that are en route, 1 for aircraft that are not currently mission-ready, but are at a base and can be made mission ready on issuance of a ReadyAircraft command and the passage of the specified time interval, and 2 for aircraft that have been attrited. The assignedTargetBE_Num field indicates the next target for which a ReleaseWeapons command has been issued for that aircraft and is set to zero after execution of that command. These are the data referenced in Section 7.1 as providing memory across controller cycles of the current mission objective.

The **[PackageConfiguration]** tag is repeated for each package configuration and contains information on the numbers and types of aircraft from which strike packages can be composed:

```
[PackageConfiguration]
packageID                     1
aircraft_type                 F2W00
number                        2
aircraft_type                 J200
number                        1
aircraft_type                 F5E10
number                        2
[endlist]
```

In other words, the controller only considers those package compositions that are specified in the list of PackageConfigurations.

The **[WeaponType]** tag is repeated for each weapon type that could be employed by the controller and should be specified for each weapon type referenced by one of the Aircraft Types (standard conventional loads). The data items contained under this tag are:

```
[WeaponType]
id                                 Mk83
size                               1000
all_weather                        no
standoff                           no
numAvailable                       4000
unitCost                           1500
release_duration                   300.
```

The **[TankerOrbit]** tag is repeated for each aerial refueling tanker location:

```
[TankerOrbit]
tankerID                           Mobil
airbase                            Andersen_AFB_Guam
center_latitude                    -1.4
center_longitude                   141.
altitude                           10000.
```

The tanker aircraft are not explicitly scheduled by the controller - they are assumed to be at the specified locations. The actual tankage demands that result from air operations is reported by the simulation. The **[Airbase]** tag is repeated for each air base at which aircraft will be deployed in the scenario.

```
[Airbase]
id                                  Gaan
latitude                           -4.66667
longitude                          144.96667
altitude                           50.
beginAvailability                  0.0
endAvailability                    999999.9
```

The beginAvailability and endAvailability data can be used to specify base closure perturbations.

The **[Target]** tag is repeated for each target:

```
[Target]
BE_number                          1
Name                               Rahaz
Region                             West_ADD
Country                            W_Cyberland
currentLatitude                    -0.929169
currentLongitude                   133.123731
currentAltitude                     0.000000
Functional_Category                IAD_C3
Graphic_References
current_damage_percent             0.00
beginAvailability                  0.
Munition_Type                      GBU31
Number_of_DMPIs                    2
```

```
Probability_of_Damage              0.7500
[endlist]
```

The simulation puts only those targets whose beginAvailability times have already been passed onto the state file.

The **[CampaignPhases]** tag contains the beginning and end times (in seconds) for each of the campaign phases defined for the scenario:

```
[CampaignPhases]
Phase                              1
Begin_Time                         0.
End_Time                           177800.
Phase                              2
Begin_Time                         177800.
End_Time                           345600.
Phase                              3
Begin_Time                         345600.
End_Time                           604800.
[endlist]
```

For each of the campaign phases that has been defined, a set of tags containing two-dimensional tables must be specified containing:

- Commanders_Priority_Table
- Valuation_Type_Table
- Pd_Threshold_Table
- Perishability_Time_Table
- Reconsitution_Time_Table

As described in Section 4, the Commanders_Priority_Table specifies relative weightings by target functional category and geographic region. An example of the format is:

```
[Commanders_Priority_Table]
campaign_phase                     1
[Table]
Priority By Region and Funcat Table
Region         West_ADD     North_ADD     South_ADD     WC_sal      Cmdrs_special
Funcat
IAD_C3         10           30            20            10          0
IAD_nonC3      5            10            5             10          0
Other_C3       10           20            5             10          0
airfields      10           20            5             10          0
LOC            1            20            1             10          0
POL            1            5             1             5           0
Man_units      1            5             1             10          0
LRarty         10           10            5             20          0
[EndTable]
```

The tabular entries in the corresponding [Valuation_Type_Table] can take on values 0, 1, or 2 for threshold or sublinear behavior, linear, and superlinear target coupling effects.

The tabular entries in the [Pd_Threshold_Table] indicate the threshold fractional damage that must be achieved on each target to enable that target to be counted toward the target damage effects for that cell of the table. The target system modeling was also described in section 4.

The [Perishability_Time_Table] and the [Reconsitution_Time_Table] tabular entries specify the time constants in seconds for the target time dynamics described in Section 4. Cells without time dynamics are indicated with a time constant of 0.

The [CostMap], [Threats], and [Table] tags contain data that specify the air defense threat model. The [CostMap] specifies the nodal grid that is used for the A* routing searches. The 30-km grid specification illustrated in Figure 4.7.2 is given by:

```
[CostMap]
CostMapCenterLongitude        139.0
CostMapCenterLatitude         0.5
CostMapAverageGridSize        30.e3
```

The [Threats] tag contains a set of parameters that specify the threat model parameters as described in Section 6.2.3. For example:

```
[Threats]
Threat_type                   SAM1
Engagement_Radius_Max         30.e3
Engagement_Radius_Max_Jam     5.e3
Engagement_Radius_Min         2.e3
Pe_No_WW                      0.50e0
Pe_WW                         0.25e0
Pk_Given_Engagement           0.01
ThreatMapCenterLatitude       -4.500000
ThreatMapCenterLongitude      137.000000
ThreatMapAverageGridSize      30000.000000
SizeOfGrid                    50
DensityOfThreatsNorm          0.24805e-10
```

specifies the parameters of the threat model, including the size of the discrete grid on which the threat density values are encoded. An example of the threat density table is given below, where the number of columns and rows has been cut down to fit within the report formatting:

```
[Table]
```

Threat Density Table

columns	1	2	26	27	28	29	30	31	32	33	34	35	36	37	38
row															
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	2	7	5	1	0	0	0
15	0	0	0	0	0	0	1	9	27	42	38	18	4	0	0
16	0	0	0	0	0	2	18	41	68	84	79	58	33	8	1
17	0	0	0	0	0	7	41	81	93	94	97	90	62	23	1
18	0	0	0	0	0	8	53	91	95	94	93	96	76	33	2
19	0	0	0	0	0	9	52	91	99	97	93	97	77	30	1
20	0	0	0	0	0	8	47	87	96	96	94	95	71	28	2
21	0	0	0	0	0	3	25	58	80	90	88	73	43	13	1
22	0	0	0	0	0	0	4	18	40	56	52	30	10	1	0

23	0	0	0	0	0	0	0	1	6	15	12	2	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Each of the table cells can take an integer value from 0 to 99, and the overall density in threats per grid cell is obtained by multiplying this integer by the DensityOfThreatsNorm parameter.

The **[MaximumMissionRisk]** tag specifies the cutoff value for estimated mission risk as a function of generic aircraft type:

```
[MaximumMissionRisk]
[Table]
Risk By Generic Aircraft Type
RiskCat                        1
ACType
F2W                            0.01
J2                             0.01
F5E                            0.01
B101                           0.01
B102                           0.005
B100                           0.01
[EndTable]
```

Missions that exceed this cutoff value are eliminated from consideration by the controller.

Finally, the last two tags, [LogCostPerFlightHour] and [UnitFlyAwayCost] specify logistical flight hour dollar costs and unit flyaway dollar costs as a function of generic aircraft type. These data are used by the controller to determine the mission that best meets the target damage objectives while at the same time exhibiting the lowest cost. The attrition cost is typically the largest component of the cost, although the cost of extremely expensive guided standoff weapons can also be noticeable.

The logistical costs per flight hour and unit flyaway cost data are listed below.

```
[LogCostPerFlightHour]
[Table]
Log Cost By Generic Aircraft Type
LogCost                        1
ACType
F2W                             2900.
J2                              3500.
F5E                             7835.
F6E                             4500.
F7D                             6500.
B101                           13156.
B102                            9401.
B100                            6501.
[EndTable]
[UnitFlyawayCost]
[Table]
Flyaway Cost By Generic Aircraft Type
UnitCost                       1
```

```
ACType
F2W                    18.8e6
J2                     26.e6
F5E                    31.1e6
F6E                    35.e6
F7D                    38.e6
B101                   283.1e6
B102                   1157.3e6
B100                   53.4e6
[EndTable]
```

7.4 Description of Scenario File Initial Data

The simulation requires the state file to initialize the simulation, with the exception that the tedious construction of aircraft instances can be generated automatically by substituting the following table for the aircraft instance enumeration in the initial data file:

```
[DeploymentTable]
[Table]
Aircraft By Type and Base Table
Base        Gaan    Mt_Hegan   Tuljeti    Darwin    Andersen_AFB_Guam
ACType
F2W          14        8          4          0               0
J2           5         2          2          0               0
F5E          18        12         6          0               0
B101         0         0          0          3               0
B102         0         0          0          0               3
B100         0         0          0          0               3
[EndTable]
```

This table indicates the number of aircraft by generic type to be initially deployed at each base. The individual aircraft instances with individual tail-number labeling are automatically generated from the information in this table.

7.5 Description of Command File

The minimum grammar necessary to describe the controller commands to the simulation include the following five items:

- ReadyAircraft
- GoToLocation
- RefuelAircraft
- ReleaseWeapons
- RecoverAircraft

The ReadyAircraft command specifies which mission load is to be applied to each aircraft instance and incurs the time delays in arming, fueling, and putting a pilot in the aircraft. The maintenance times from a fresh start and from turning around from a previous mission recovery are also specified.

The GoToLocation command is the only command that specifies a transit activity. The simple model is a great circle route at constant speed from the current location to the specified location with detailed routing being constructed from sequences of GoToLocation commands. All other activities require that the aircraft be at particular locations in order to complete the activity successfully and are usually preceded by one or more GoToLocation activities.

RefuelAircraft activity requires proximity to a tanker location and passage of an activity duration to reinitialize the aircraft fuel endurance remaining.

The ReleaseWeapons activity specifies what weapons are released onto what targets and also incurs an activity duration. If the aircraft is not within weapon range from the release location for a target, no release is actually implemented.

The RecoverAircraft command requires that the aircraft be within proximity to a base to be implemented and also incurs a time delay.

Activities are executed in sequence, with refueling and weapon release timing out after the specified duration regardless of outcome. The earliest start time is specified for each activity so that an aircraft arriving at a location before the earliest start time for the subsequent activity performs a loiter maneuver at that location. The trajectory during the time duration of refueling, releasing weapons, and recovering the aircraft is also expressed as a loiter. For fixed-wing aircraft, the loiter is implemented analytically as a 1.25-g banked turn from a slowed down speed of 385 kn with departure from the loiter orbit (approximately 5-km radius) at whatever location corresponds to the commencement of the succeeding activity. The intent of this feature is to provide some visibility into the 4D flight path during the execution of nontransit activities without belaboring the intrinsically simple model for flight path propagation that is appropriate for high-level planning.

A short example of tags associated with a command file is given below:

```
[AircraftPlan]
tail_number               MO0423

activityType              ReadyAircraft
activityID                1
earliestStart             0
latestEnd                 0
minDuration               0
mission_configuration     F5E06
role                      striker

activityType              GoToLocation
activityID                2
earliestStart             5157.36
latestEnd                 7364.67
minDuration               2207.31
latitude                  -2.93393
longitude                 140.701
altitude                  0

activityType              ReleaseWeapons
activityID                34
earliestStart             3169.79
latestEnd                 3169.79
minDuration               300.
beNumber                  58
munitionType              GBU31
```

```
number                          4

activityType                    GoToLocation
activityID                      2
earliestStart                   5157.36
latestEnd                       7364.67
minDuration                     2207.31
latitude                        -1.93393
longitude                       143.701
altitude                        0

activityType                     RefuelAircraft
activityID                      1
earliestStart                   0
latestEnd                       0
minDuration                     0

activityType                    GoToLocation
activityID                      2
earliestStart                   5157.36
latestEnd                       7364.67
minDuration                     2207.31
latitude                        -2.53393
longitude                       144.701
altitude                        0

activityType                    RecoverAircraft
activityID                      356
earliestStart                   25007.6
latestEnd                       25007.6
minDuration                     0
```

After the AircraftPlan tag, the next nonblank record is the aircraft tail_number, the unique identifier for that aircraft instance. There is then a sequence of activities for that aircraft that may span several days worth of sorties that is terminated when the next AircraftPlan tag is read or the file is terminated. The first activity is usually a ReadyAircraft activity if the aircraft is at a base. If it is en route, the first activity of a diversion for a changed plan will be a GoToLocation activity. The data associated with each activity depend on the type of the activity. Multiple weapon releases on the same or different targets are indicated by repetition of the ReleaseWeapon activity.

The latest end time for each activity can be used to indicate the election to abort or skip activities that cannot be completed by the specified time. Normal activity completion occurs at the actual start time plus the specified minimum duration, except the GoToLocation activity that completes on arrival at the specified location. All times are indicated in units of seconds, with absolute time referenced to an arbitrary zero time for the beginning of the scenario. All geographic coordinates are specified in units of decimal degrees for latitude and longitude, and meters for altitude.

8 Experiments

8.1 Discussion of Experimental Plans, Metrics, Objectives, and Results

This section summarizes the experiments that we have performed in our research program. The experiments have been defined to test hypotheses about the performance of the air operations enterprise controller whose design was described in Section 5. The experiments are performed in the context of the simulation that was described in Section 6. A baseline scenario and a set of performance metrics are described in Section 8.2. The application of the controller against the baseline scenario establishes a basis of comparison for our other experimental results. Our experiments have been phased to allow us to first validate the closed-loop functionality of our controller and to progress in experiment complexity to testing the range of experimental hypotheses defined in Sections 8.4 and 8.5.

The experiments are organized into two sets: (1) those that evaluate the performance of the controller when the loop is closed at fixed intervals and replanning is performed on a cyclic basis, and (2) those for which the loop is closed aperiodically to reflect information coming back from the plant (the air operations enterprise) as events occur, with replanning occurring as a function of the nature of those events (e.g., an attritted aircraft or new targets defined by the Commander). Our principal hypothesis is that loop closure and replanning at higher rates (or as relevant events occur) can provide substantial improvement in performance of the overall enterprise.

8.1.1 Cyclic vs Event-Based Loop Closure Experiments

The feedback between our simulation of the plant and the controller is potentially relevant when there are events or information that are not completely revealed at start time. This includes the outcome of stochastic attrition modeling. It also includes the disclosure of targets over time and the disclosure of discrepancies between actual and forecast target damage states as a result of deliberately introduced modeling errors.

Our earliest design and implementation of the controller architecture allowed for plant feedback and air operations replanning at fixed intervals only. This early design allowed us to model the approach that is currently taken in air operations planning. That is, a completely new air operations plan is regenerated every 24 h. The first set of experiments discussed below were based on this constrained approach to loop closure and provided insight into improvements that could be achieved if the entire air operations plan were regenerated at higher rates.

The second set of experiments has been performed with the more advanced controller architecture that allows for local and aperiodic replanning. Depending on the nature of the events that are fed back to the controller, a local replan may suffice. Thus, the entire air operations plan need not be updated to accommodate an event that has only local (to a subproblem) impact.

8.2 Experimental Setup

Our experiments were executed in the context of our own plant simulation environment described in Section 6. This section discusses how experiments were conducted, how results were compared, and the cost and performance models that were applied.

At the lowest resolution view, an attack plan produces a result that has characteristics of interest, including:

1. *Aggregate damage levels* visited on the target set.
2. The *time* to achieve a specified aggregate damage level.
3. The *cost* to achieve that result.

The latter includes operating costs, ordnance costs, and the cost of aircraft attrition. Of secondary interest is the time to derive the attack plan, a time that will become of interest only when it becomes substantial with respect to other characteristic times in the domain. Although computation times are of practical importance in an operational system, we are not addressing them directly here. However, we do compare computation time differences in some of our experimental results.

Each simulation in support of an experiment involves defining:

- The underlying **scenario** data.
- The number of **Monte Carlo** repetitions.
- The **stopping rules** for the simulation.

We address each of these in the following subsections.

8.2.1 Baseline Scenario Key Elements

Our baseline scenario is derived from that described in detail in Section 6. The key information elements that characterize a scenario in our simulation model are:

- Threat laydown.
- Target laydown.
- Target time sensitivity.
- Bases/aircraft.
- Weapons and weapons effectiveness.
- Tankers and assembly points.
- Commander's Intent.
- Mission risk aversion.
- Cost to replan a mission that is already in progress in the air.
- Loop closure cycle time interval.

8.2.2 Monte Carlo

Monte Carlo sampling and averaging is relevant for those processes that are stochastic by nature and also for those processes where there is a sensitivity to parameters whose operational values are unknown or contain a high degree of variability. In the JFACC project, we apply Monte Carlo to model the stochastic process of aircraft attrition with respect to threat encounters in all simulations. We also selectively employ Monte Carlo sampling to assess the variability to details in the Commander's Input Priorities (see Landscape experiment). Although weapon effects on targets has a stochastic element, we assume that conservative weaponeering is employed and yields specified target damage levels to a high degree of confidence taking into account target location error, duds and weapon release errors, justifying the deterministic modeling of the target damage process.

8.3 Metrics

The primary metrics that we employ to evaluate the experiments are those associated with accomplishing the Commander's Intent (see Section 4.1). Those metrics include the aggregate value of target destruction by: (1) target category, (2) operational geographic region, (3) campaign phase (time), as well as (4) time to achieve levels of fractional destruction along these same dimensions.

On the cost side, the attrition of aircraft and the cost of utilization of munitions and mission support resources are also logged and included as an element of the evaluation for each experiment. In addition to cost and plan value, we also evaluate the performance of our closed-loop controller in the context of "plan stability." Here, plan stability is defined a measure of how plans change each time the loop is closed and new plans are developed. From a human factors perspective, it is unacceptable to have frequent, significant changes in strike plans for individual aircraft, especially when they are already en route to a target. Our plan stability metric is a measure of the number of those changes.

8.3.1 Aggregate Target Damage Metric

Figure 8.3-1 is a representative illustration of the accumulation of the aggregate damage metric accrued over the course of a 7-day campaign. As described in the annotation in the figure, there is both a low- and high-frequency structure to the curve. The high-frequency structure reflects that fact that the aggregate damage is normalized by the total number of targets that have been disclosed (i.e., identified by ISR and put on the list of potential targets by the Commander). The damage curve rises as the Commander's Intent for attacked targets is achieved and falls as the denominator increases when new targets are disclosed. In this regard, the controller that acts on disclosed targets on a 24-h cycle is at a significant disadvantage.

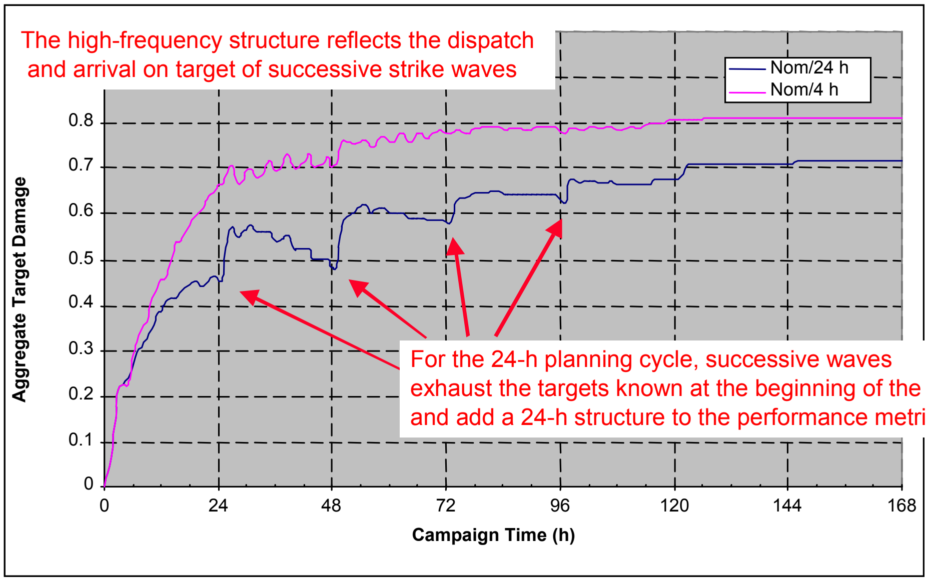


Figure 8.3-1. Aggregate Damage Metric for 4- and 24-h Controller Cycles.

There is a 4-h period for disclosure of new targets that shows up in the damage metric. The aggregate damage is normalized so that new (undamaged) targets that are added to the target list decrease the metric. There is a discontinuity in Commander's Intent (the weighting for the metric) at 48 h and at 96 h corresponding to shifting emphasis with transitions between campaign phases.

8.3.2 Total Time to Achieve Campaign Objectives

An important metric is the time that a given controller takes to achieve a specified level of aggregate target damage. Indeed, as the aggregate target damage is a measure of the level of achievement of the Commander's Intent, an important measure of performance is the amount of time required to achieve that level. This can be viewed as the amount of time it takes to accomplish the goals of a given campaign phase, allowing the Commander to move forward to the next phase.

For example, transitioning from Phase 1 to Phase 2 of a campaign may specify a 75% reduction in armor and field artillery units. This objective is desired at a specified time as coordinated in the larger plans of the theater or task force Commander. Given a set of resources (deployed aircraft) to accomplish this objective, the JFACC desires plans that achieve this objective target damage in the required time and with acceptable costs. Under normal procedures, a set of alternative options or plans is considered, and the JFACC will choose from among the options provided by the JFACC planning staff.

The outcome from the staff that provides a comparison among those plans might be of the form of Figure 8.3-2.

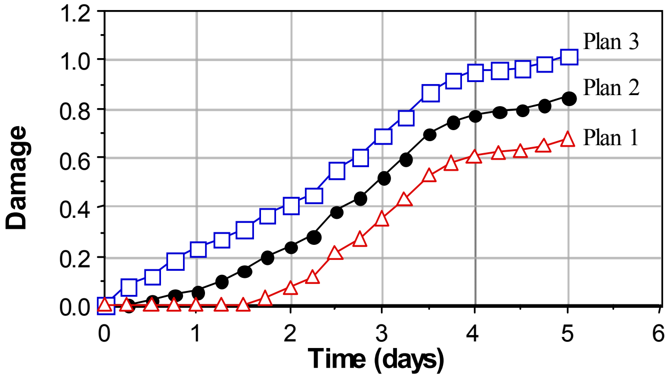


Figure 8.3-2. Damage Metric vs Time for Several Plan Options.

Since the guidance from the theater commander to the JFACC is a specified damage level objective, what is perhaps more relevant to the JFACC is a presentation of plan options in the format of Figure 8.3-3.

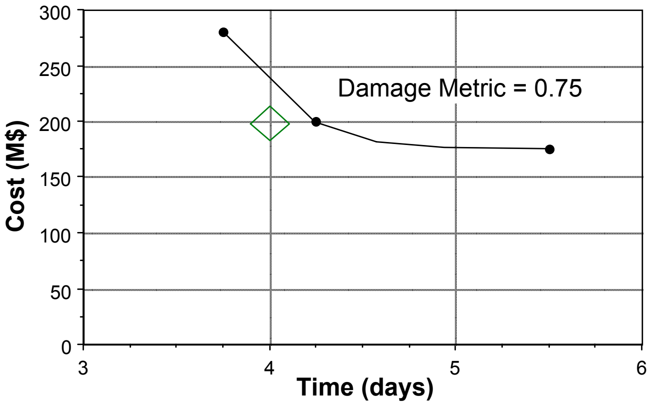


Figure 8.3-3. Cost vs Time of Plan Options that Achieve Specified Objectives.

Figure 8.3-3 shows the plan options in the space of the cost and time to achieve the specified objective. Here we also include the Commander's bogey in terms of desired time and campaign cost to achieve that level of objective and is indicated by the diamond around 4 days and $200M. The JFACC might relax the cost bogey and direct the staff to tweak the plan to achieve the objectives at the 4-day phase transition time negotiated with the theater Commander; he might try to negotiate a 4.25-day phase transition, or he might allocate or ask for more resources to accomplish the objectives within the desired time and cost bogies. What is noteworthy is that the planning constraint for the JFACC is the specified damage level and that the plan options trade cost versus time to achieve that objective. If none of the options are acceptable, then it is conceivable that the damage objectives could be relaxed in negotiation with the theater command level, a process that is outside our scope of interest.

8.3.3 Cost Model and Risk Limits

The cost includes the cost of attrition, the cost of operations, and the cost of munitions. Although the JFACC ***cost*** considerations are primarily concerned with losing aircraft and air crews, the formulation in terms of total dollar cost reflects secondary but very important logistical considerations. It also reflects the real calculus that losing a $1 billion B-2A is in a different category from losing a $31 million F-15E.

Note that there is never any desire to try to compare target value with our own costs. This is always a problematic issue that is best avoided. If targets are on the approved target list, they are intended to be reduced subject to constraints on our willingness to assume risk in attacking those targets. Comparing the experiments at a specified damage metric as opposed to a fixed scenario time will avoid the issue of comparing disparate damage levels and will also emphasize differences between results that are most meaningful to the JFACC.

8.3.3.1 Cost Model

The cost model includes the cost of:

- Operations.
- Munitions.
- Attrition.

using data from Air Force cost estimating documents (AFI 65-503). The cost of operations includes fuel and maintenance items that are proportional to the number of flight hours. The supporting SEAD aircraft are explicitly accounted for in our models, although the supporting missions for counter air combat air patrol aircraft and tanker operations are not explicitly modeled in a way that we can estimate and allocate those costs. Hence, those costs are not included.

The logistical costs per flight hour and unit flyaway cost data are listed in the interface section.

The mission cost is estimated in the controller by applying the router-derived probability of attrition multiplied by the unit flyaway cost and adding the router-supplied mission time duration times the logistical cost per flight hour. These costs, and the cost of ordnance expended, are contained in the state file. The simulation will add an attrition cost only when an attrition event is sampled.

8.3.3.2 Risk Limit – Risk Aversion

A table of maximum acceptable mission risks is added to the state file for using the controller to decide whether to accept a mission depending on the probability of attrition as returned by the router. Risk values were derived by assessing the contribution of attrition cost to the overall mission cost in the context of reasonable scaling of peacetime attrition rates for the various aircraft types. For a heavy bomber, the peacetime attrition rate for a 5-h sortie is on the order of 1 in 40,000 sorties. If we allow a factor of 10 escalation for employment of the B-2A in combat, the attrition cost for a 1 in 4000 sorties attrition loss could be considered consistent with the other mission cost elements. Similarly, for tactical strikers with higher peacetime attrition rates but much lower costs, the upper limit of acceptable combat attrition costs may vary from 1 in 400 sorties to 1 in 250 sorties to be consistent with overall mission cost structure.

Aircraft Type	Maximum Mission Risk
F2W	.00400
J2	.00250
F5E	.00250
F6E	.00250
F7D	.00250
B101	.00100
B102	.00025
B100	.00125

8.3.4 Auxiliary Metrics

Auxiliary metrics include:

- Plan stability metric.
- Damage by target type.
- Total number of sorties.
- Weapons used.
- Fuel used.
- Aircraft attrition.

8.4 Cyclic Loop Closure Experiments

Table 8.4-1 summarizes the set of cyclic loop closure experiments that were performed. Table 8.4-2 summarizes the set of event-based controller experiments. The baseline scenario is the Cyberland Scenario with 313 targets disclosed over five days of campaign as described in Section 6.

Table 8.4-1. Cyclic Controller Experiment Summary.

Closed-Loop Cyclic Experiment	Controller Configuration	Scenario Variation	Experimental Objective
Problem Decomposition	Undecomposed and Decomposed	Baseline	Determine the computational benefits of decomposing the problem
Loop Closure Rate	4-h and 24-h loop closure intervals	Baseline	Determine the performance benefits of higher rates of feedback
Sensitivity to Modeling Errors			
- *Weapons Effectiveness*	4-h loop closure interval w/ effectiveness of weapons underestimated in the controller's model	Weapons effectiveness reduced by a factor of 2 - 5	Determine the robustness of the closed-loop controller to errors in weapons effectiveness modeling
- *Air Defense Effectiveness*	4-h loop closure interval w/ effectiveness of enemy air defense underestimated in the controller's model	Lethality of enemy air defense increased by a factor of 2	Determine the robustness of the closed-loop controller to errors in enemy air defense effectiveness modeling
Sensitivity to Unexpected Scenario Changes			
- *Commander's Intent*	4-h loop closure interval	Unexpected change in target values by category and region due to change in Commander's Intent	Determine agility of the closed-loop controller in responding to target value changes
- *Base Closures*	4-h loop closure interval	Unexpected closure of an air base due to poor weather conditions	Determine agility of closed-loop controller to reconfigure air operations plans in the face of base closure
All UCAV Fleet	4-h loop closure with reduced sensitivity to risk and reduced cost of retasking aircraft in flight	All aircraft unmanned	Determine the improvement in performance and increased attrition when all aircraft are unmanned
Time-Sensitive Targets	4- and 24-h loop closure intervals	Variations include: all TSTs, no TSTs, all TSTs and no air defense threat	Determine ability of the closed-loop controller to plan and execute missions for various levels of TSTs

Table 8.4-2. Event-Based Controller Experiment Summary.

Event-Based Controller Experiment	Controller Configuration	Scenario Variation	Experimental Objective
Comparison to Cycle	Event-based/4 hour cyclic	Baseline	Determine performance benefits of the controller reacting (replanning) as events occur
Increased Air Defense Threat	Event-based with negotiation /4-h cyclic w/o negotiation (both controllers correctly model increased threat)	Air defense lethality increased by a factor of 5	Determine improvement due to more timely response to attritted aircraft in a package
All UCAV Fleet	Event-based/4-h cyclic both with reduced sensitivity to risk and reduced cost of retasking aircraft in flight	All aircraft unmanned	Determine ability of the closed-loop controller to plan and execute missions for various levels of TSTs

All the experimental data are included in Figure 8.4-1.

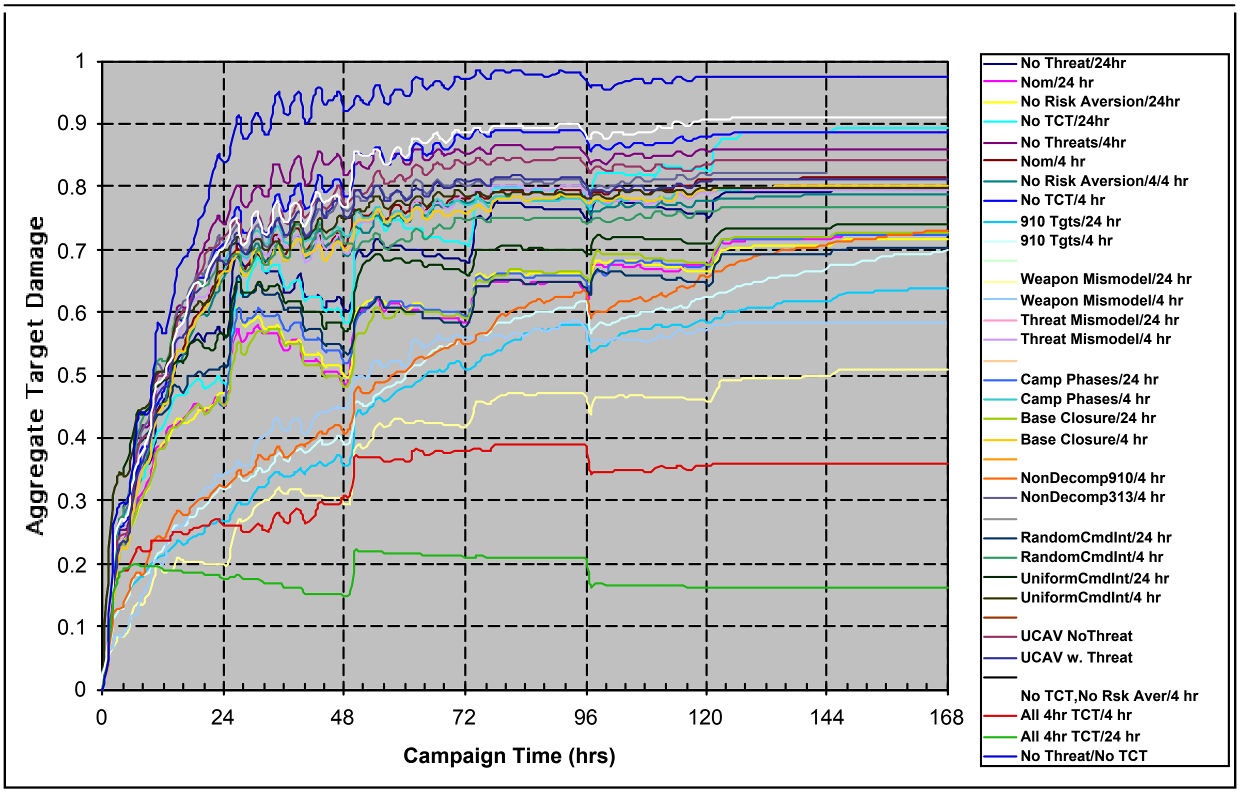


Figure 8.4-1. All the Experiments.

8.4.1 Problem Decomposition

In the experiment descriptions that follow, the heuristic planner is employed in both an undecomposed and a decomposed fashion. The performance of the planner is measured by evaluating the aggregate target damage metric.

Hypothesis: Decomposing the air operations optimization problem substantially decreases the computational, burden, provides solutions to subproblems that are easily comprehendible to human operators, and results in near-optimal solutions.

Scenario: We will compare solutions to air operations tasking problems for decomposed and undecomposed formulations. The comparison will be based on: (1) computational effort and (2) human evaluation of solutions. The experiment will be executed in a relatively simple scenario with a loop closure rate of once per hour.

Results: The set of metrics that will be employed to evaluate these experiments includes the aggregate value of target destruction by category, region, and time, and fractional destruction along these same dimensions. These metrics correspond to Level-3 performance—with top-level military relevance to the overall air operations mission.

From Figure 8.4-2, it is clear that decomposition reduces computation time significantly. Reduction in computation time due to decomposition is greater if subproblems are solved in parallel. However, the effect of parallel processing is not as great as we would like because, in the decomposition by air base, one or two subproblems do the "heavy lifting." Decomposing so that each subproblem has roughly the same resources and targets should maximize the computational benefit of decomposition. In this scenario, the improvement is by a factor of 5. The computation time is normalized by the time-to-plan for the full, undecomposed problem (910 targets).

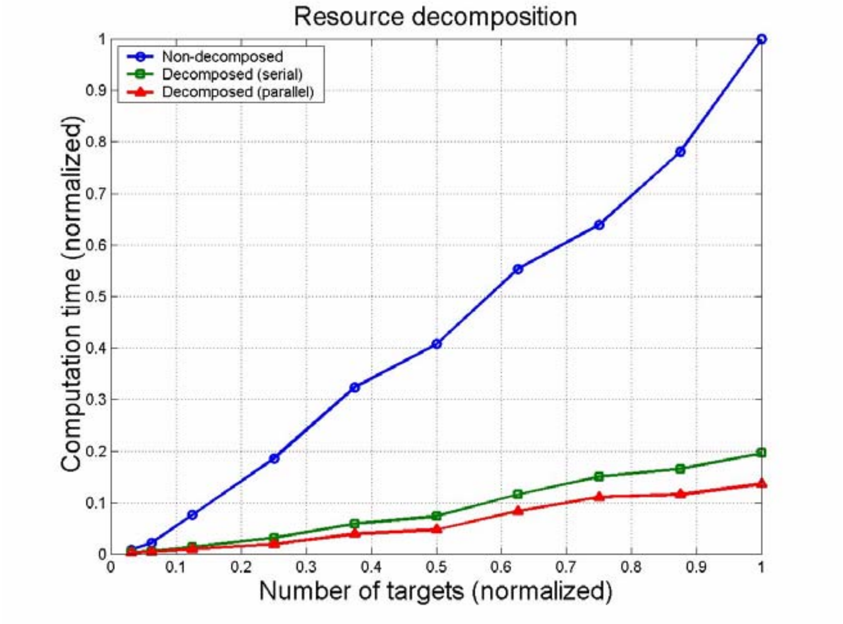


Figure 8.4-2. Impact of Problem Decomposition.

While the decomposition lowers the required computation time, it can also lower the quality of the solution—the solution algorithms are no longer allowed to consider inputs over the entire problem space, potentially limiting the effectiveness of the solver. To examine this situation, in Figure 8.4-3 and Tables 8.4-3 and 8.4-4, we compare the normalized target damage metric of the undecomposed heuristic solver and the decomposed heuristic solver. We ran two cases: the baseline 313-target case and the extended 910-target case. From the figure, it is clear that the decomposed heuristic solver achieves a lower level of target damage, but it is encouraging to note that the difference is not large.

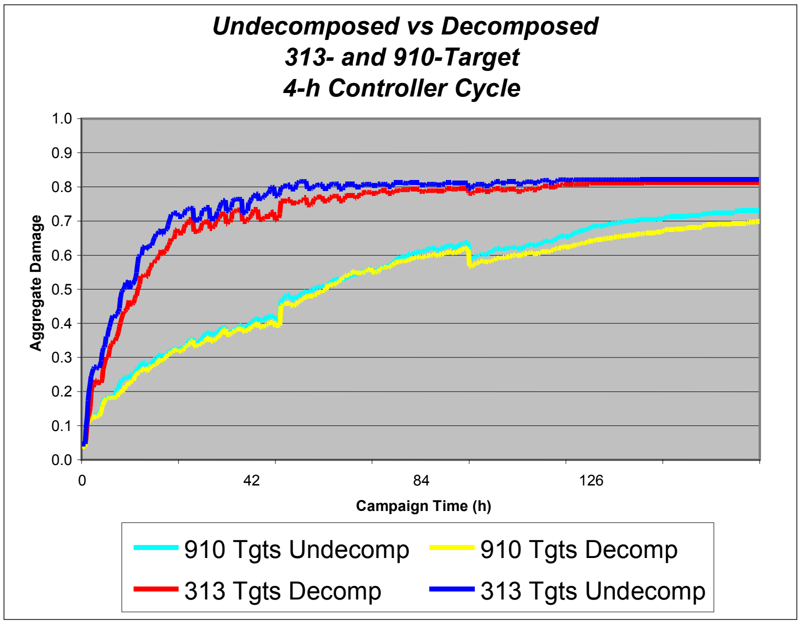


Figure 8.4-3. Undecomposed Heuristic Solver vs Decomposed Heuristic Solver.

Table 8.4-3. 313-Target Scenario, 4-h Cycle.

Day Number	1	2	3	4	5	6	7
Undecomposed	0.714	0.774	0.804	0.797	0.821	0.822	0.822
Decomposed	0.665	0.703	0.780	0.782	0.808	0.813	0.813

Table 8.4-4. 910-Target Scenario, 4-h Cycle.

Day Number	1	2	3	4	5	6	7
Undecomposed	0.322	0.409	0.549	0.633	0.654	0.706	0.730
Decomposed	0.318	0.391	0.550	0.617	0.624	0.670	0.699

Conclusions: Decomposing the problem into subproblems substantially reduces the computational effort required to solve the problem. Furthermore, the gain in computation time comes at a small cost in terms of solution quality.

8.4.2 Loop Closure Rate

In the experimental results that follow, we focus on the difference in performance of closing the air operations planning and control loop at a low rate (24-h cycle) and a relatively high rate (4-h cycle). Four-hour and shorter cycles are made possible by automating the plan generation process as described earlier and improved battlefield information dissemination, providing the required higher rate of feedback.

Hypothesis: Significant improvements in air operations effectiveness are gained through higher (than 24 h) rates of loop closure.

Scenario: We will establish two baselines: one for which the controller loop is closed every 24 h, and one wherein the loop is closed every 4 h. The first case is intended to roughly represent performance levels for the approach that is taken today. The second case represents what we anticipate would be the highest rate that is feasible given that there will necessarily be humans in the loop. These baseline experiments will investigate the differences between the 4-h cycle and the 24-h cycle for both a 313-target case and a 910-target case. All the experiments in this first set will have attrition and weapons effectiveness plant models consistent with those employed in the controllers in making air operations planning and situation assessment decisions.

Results: The set of metrics that will be employed to evaluate this experiment includes the aggregate value of target destruction by category, region, and time along these same dimensions. These metrics all correspond to Level-3 performance—with top-level military relevance to the overall air operations mission.

The results of our experiments show that there is a general overall increase in performance that can be achieved by shortening the loop closure rate. Figure 8.4-4 and Tables 8.4-5 and 8.4-6 show that the 4-h cycle provides better performance in the terms of aggregate damage achieved over time for both scenarios, with the greatest difference being for the 313-target case for the 7-day campaign time.

The high-frequency ripple in the results is a combination of new target disclosure from ISR resources at 4-h intervals, reducing the (normalized) aggregate damage metric and the damage visited on targets by successive waves of strike packages. Note that for the 24-h planning cycle, successive waves of package sorties (see Figure 8.4-5) exhaust the targets known at the beginning of the day and add a 24-h structure to the performance metric.

The 4-h controller cycle exhibits a more level use of strike package resources over time and results in striking nearly all the known targets that are feasible by the end of the fifth day. Since the attrition of strike packages is simulated probabilistically, we have made a series of Monte Carlo experimental runs (see Figure 8.4-6) to determine the dispersion of results.

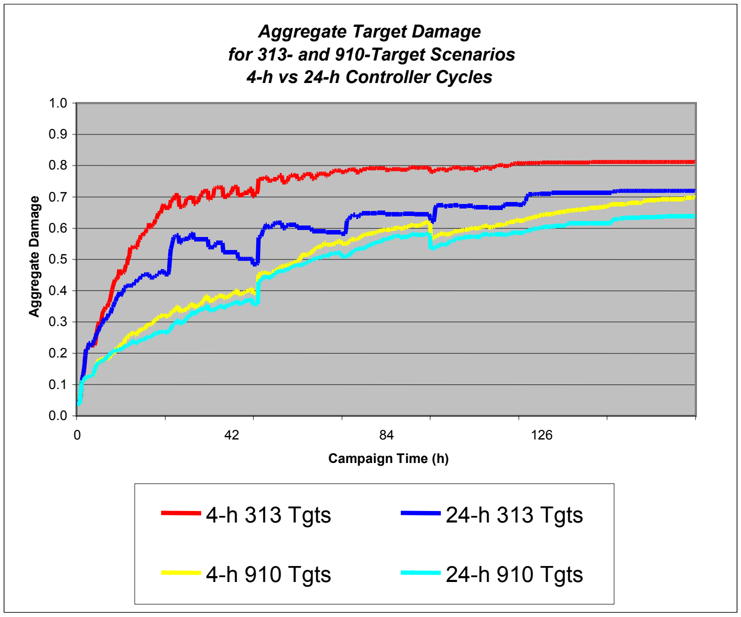


Figure 8.4-4. Baseline 313- and 910-Target Scenarios for 4-h and 24-h Controller Cycles.

Table 8.4-5. 313-Target Scenario.

Day Number	1	2	3	4	5	6	7
4-h Cycle	0.665	0.703	0.780	0.782	0.808	0.813	0.813
24-h Cycle	0.452	0.485	0.582	0.634	0.675	0.713	0.720

Table 8.4-6. 910-Target Scenario.

Day Number	1	2	3	4	5	6	7
4-h Cycle	0.318	0.391	0.55	0.617	0.624	0.670	0.699
24-h Cycle	0.268	0.355	0.509	0.578	0.585	0.616	0.638

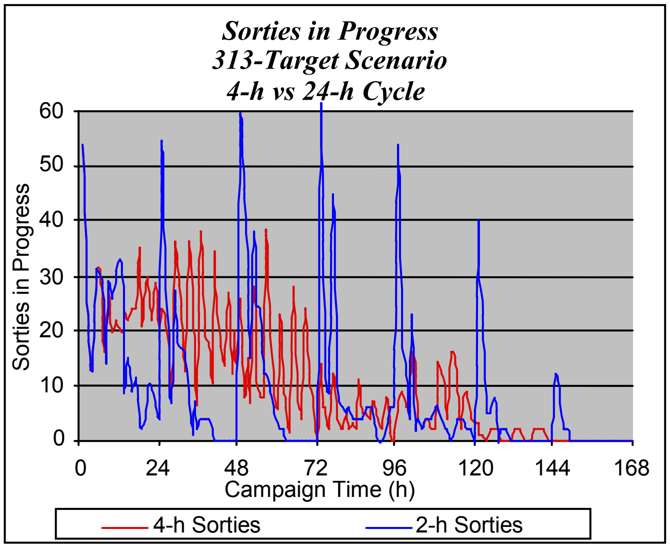


Figure 8.4-5. Sorties Generated for 4-h and 24-h Controller Cycles.

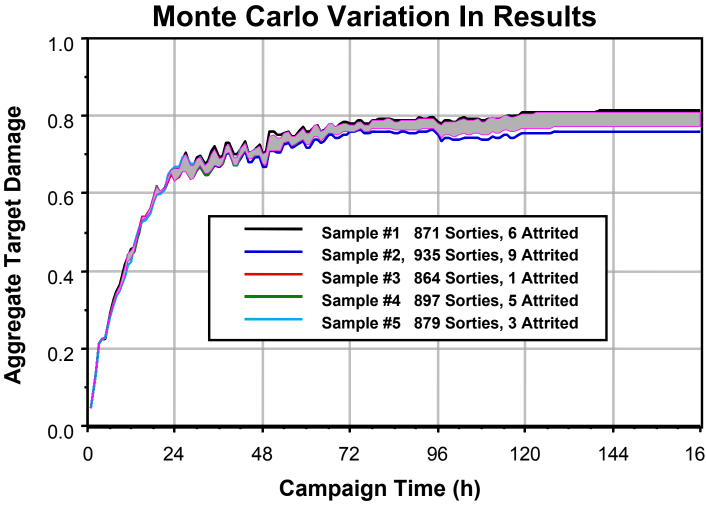


Figure 8.4-6. Dispersion of Aggregate Damage for Monte Carlo Trials of 313-Target Scenario.

A significant result that we have observed through our experimentation is that the ability to close the loop at higher rates shortens the total time required to achieve campaign objectives. To quantify, we have performed a least-squares fit through the aggregate target damage vs time curve, and extracted the difference in time to achieve different levels of aggregate target damage using the 4-h vs 24-h loop closure rate. The results are shown in Figure 8.4-7. The benefit for the baseline 313-target case ranges from 12 to 48 h for types ranges of target damage objectives, a substantial gain. For the 910-target scenario, the benefits ranged from 8 to 24 h. Figure 8.4-7 also shows that this benefit accrued in the cases of modeling error as well as for dynamic scenario changes. For the case of weaponeering error, the benefit could be even larger.

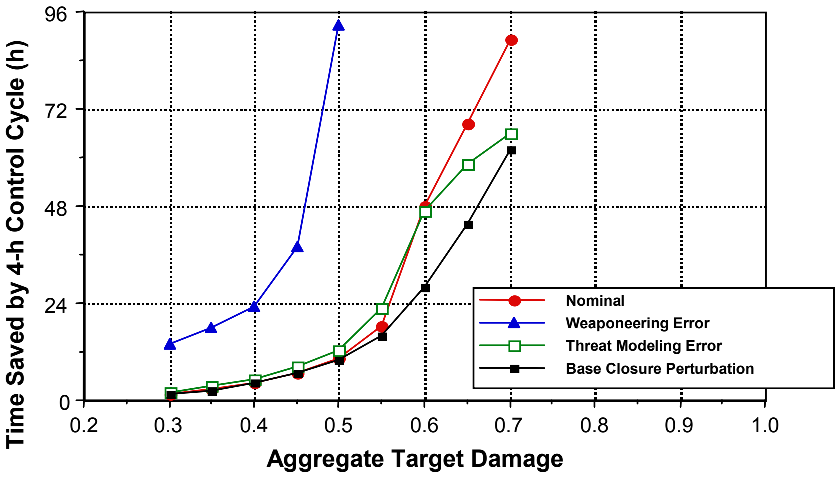


Figure 8.4-7. Campaign Time Saved by 4-h Cycle over 24-h Cycle.

Conclusions: There has been a long-standing advocacy by those in the command and control community for closing the command and control feedback loop at ever shorter intervals. This advocacy relates to a *rule of thumb* in control system design: to achieve good system performance and maintain robustness, the total time lag for each control cycle attributable to: (1) measuring and conditioning feedback signals, (2) information transfer, and (3) control law computation should be one-fifth to one-tenth the time constant of the fastest dominant mode of the plant to be controlled. Applying that rule of thumb to the control of military air operations, our goal should be command and control cycle times that are five to ten times shorter than those of our adversaries. Although the differences between the nature of the plant to be controlled by a traditional closed-loop controller and that to be controlled by a military command and control system are significant, there are obvious advantages in an ability to plan, execute, and replan many times faster than one's enemy.

The work reported in this report is one of the first instances where the benefits of higher rate loop closure have been quantified for a complex enterprise command, and control application such as coordinating air attack operations spanning the air operations enterprise from JFACC level to the strike package level. Our experimental results show that the benefits are substantial and that they accrue even in the face of the types of model discrepancies that are to be expected in such applications.

8.4.3 Sensitivity to Modeling Errors

In the following section, we focus on the response of the controller to uncertainties in the models used for blue aircraft attrition and blue weapons effectiveness, the effect of removing access to one of the blue air bases, and the effect of both large and small changes in the Blue Commander's guidance.

8.4.3.1 Weapon Effectiveness

Hypothesis: Closing the loop improves controller robustness to uncertainties in the models used for blue aircraft attrition and blue weapons effectiveness.

Scenario: We ran experimental trials wherein either or both of the weapons effectiveness models and the aircraft attrition models employed in the plant are inconsistent with those employed in the optimization models that are employed in the development of air operations plans.

Results: The set of metrics that were employed to evaluate these experiments includes the aggregate value of target destruction by category, region, and time and fractional destruction along these same dimensions. Aggregate results are shown in Figure 8.4-8. Costs relating to the attrition of aircraft and the utilization munitions will be included in the evaluation as well. These metrics all correspond to Level-3 performance—with top-level military relevance to the overall air operations mission.

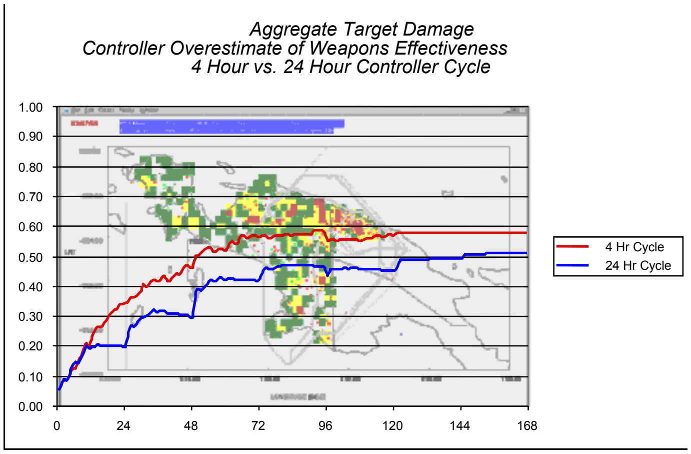


Figure 8.4-8. Sensitivity to Modeling Errors: Weapons Effectiveness.

Conclusions: Compared with the baseline case, this perturbation showed a small increase in attrition and a slightly better aggregate damage performance. The controller operating at the 4-h cycle is able to respond more quickly to BDA reports and reattack high-value targets. Higher loop closure rates would be required to respond quickly enough to impact TSTs.

8.4.3.2 Enemy Air Defense Effectiveness

Hypothesis: Varying IADs effectiveness, noise in the estimates of the state fed back to the controller, and the interpretation of the Commander's Intent will eventually result in poor controller performance.

Scenario: For this experiment, we investigate the sensitivity rather than the absolute value of the performance of the controller to the variations in IADs effectiveness, noise in the estimates of the state fed back to the controller, and the interpretation of the Commander's Intent. For the IADs effectiveness, we will determine sensitivity to a value/risk metric. That is, we will determine at what point the risk of attacking against a very strong air defense overwhelms the value of the targets prosecuted successfully.

Results: The set of metrics that were employed to evaluate these experiments includes the aggregate value of target destruction by category, region, and time and fractional destruction along these same dimensions. Aggregate results are shown in Figure 8.4-9. Costs relating to the attrition of aircraft and the utilization munitions will be included in the evaluation as well. These metrics all correspond to Level-3 performance—with top-level military relevance to the overall air operations mission.

Conclusions: Compared with the baseline case, the attrition increased as expected, but the controller maintained the ability to achieve baseline target damage metrics. The faster feedback loop enabled significantly shorter times to achieve specified damage levels.

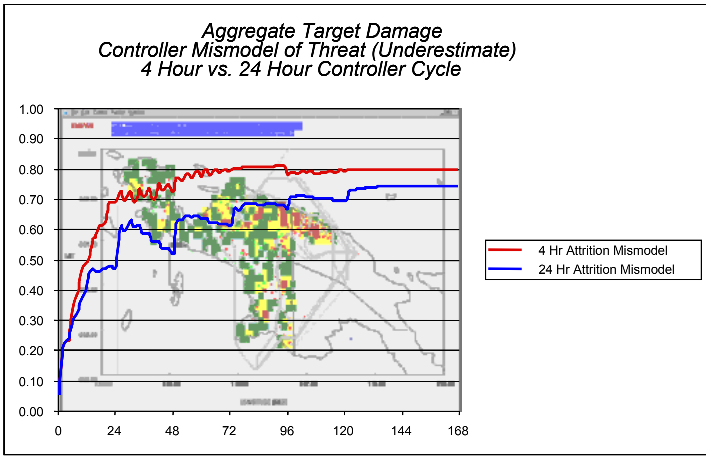


Figure 8.4-9. Sensitivity to Modeling Errors: Attrition Rate.

8.4.4 Sensitivity to Unexpected Scenario Changes

8.4.4.1 Change of Commander's Intent

Hypothesis: Closing the loop improves controller agility in response to changes in the priority/importance of existing targets (i.e., changes in Commander's intent).

Scenario: We executed experimental trials for scenarios that reflect significant changes in campaign objectives. For instance, the focus of campaign operations may shift from one geographic region to another, thus lowering the values of targets in the old region of focus and increasing the values of targets in the new region of focus. Alternatively, the importance to the campaign of a specific subset of targets within a region may be increased, and that importance is reflected in an increase in their target value. We will analyze the ability of the system to reconfigure and retask resources to accommodate the changing values.

Results: The set of metrics that was employed to evaluate these experiments includes the aggregate value of target destruction. Results are shown in Figures 8.4-10, 8.4-11 and Table 8.4-7. The metrics all correspond to Level-3 performance—with top-level military relevance to the overall air operations mission. The overall evaluation will be based on the ability of the system to reconfigure and retask resources to accommodate the changing laydown/importance of targets. The improved agility achieved by the 4-hour cycle relative to the 24-hour cycle is dramatically illustrated in Figure 8.4-11.

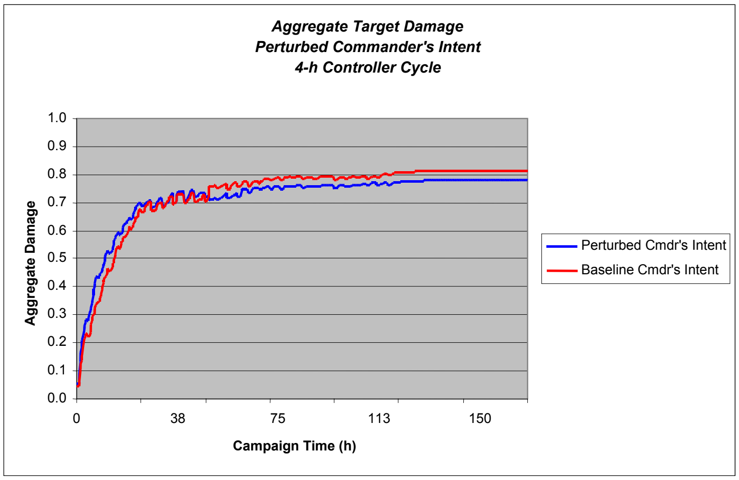


Figure 8.4-10. Sensitivity to Variations in Commander's Intent.

Table 8.4-7. Target Scenario, Perturbed Commander's Intent.

Day Number	1	2	3	4	5	6	7
Baseline Cmdr's Intent	0.665	0.703	0.780	0.782	0.808	0.813	0.813
Perturbed Cmdr's Intent	0.687	0.719	0.750	0.750	0.771	0.779	0.779

Conclusions: The controller operating at the 4-h cycle responds immediately to the changed campaign objectives. Both near-term and long term improvements in achieveing Commander's Intent was achieved. Although the controller is responsive to Commander's Intent, it is not overly sensitive to small variations in those numbers.

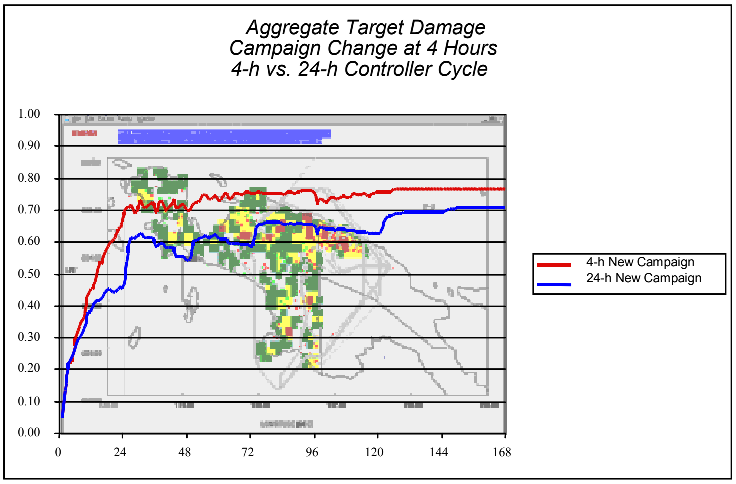


Figure 8.4-11. Dynamic Scenario Changes: Campaign Phase Change at 4 h.

8.4.4.2 Base Closures

Hypothesis: Closing the loop improves the controller's agility in response to large perturbations, such as temporary base closure.

Scenario: We executed experimental trials for scenarios that contained an unexpected and temporary closure of the second most important East Cyberland air base (Mt. Hegan). All bases were initially operational. At the start of the second day of operations, Mt. Hegan air base and all landed aircraft were unavailable to support operations for a period of 2 days. This scenario tested the ability of the controller to rapidly reconfigure air tasking in the face of major perturbations.

Results: The set of metrics that were employed to evaluate these experiments includes the aggregate value of target destruction by category, region, and time and fractional destruction along these same dimensions. Aggregate results are shown in Figure 8.4-12. Costs relating to the attrition of aircraft and the utilization munitions will be included in the evaluation as well. These metrics all correspond to Level-3 performance—with top-level military relevance to the overall air operations mission.

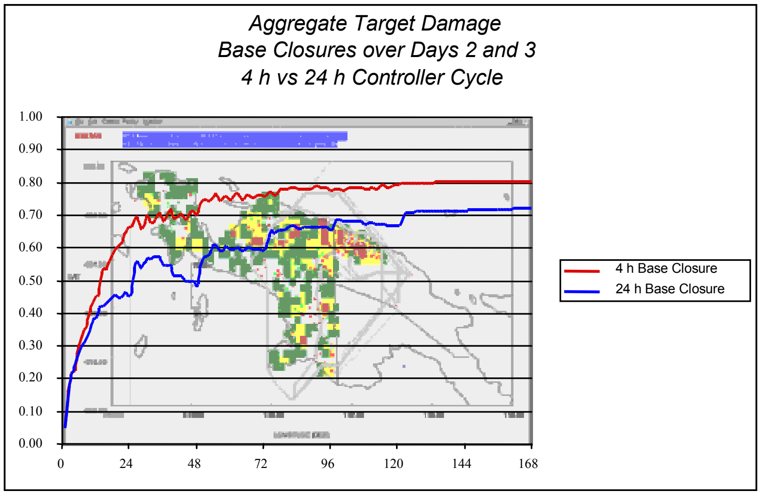


Figure 8.4-12 Dynamic Scenario Changes: Base Closure.

Conclusions: The controller operating at the 4-h cycle makes more effective use of limited resources available during periods of base closure. Tasking was reconfigured rapidly to maintain a high sortie rate from remaining bases, and unperturbed target damage levels were achieved with small delys relative to the baseline case.

8.4.5 Scenario Variations

8.4.5.1 All UCAV Fleet

Hypothesis: When dealing with an all-UCAV fleet, we were able to remove the inherent cost of subjecting aircraft and pilots to high risk, the goal being to permit the controller to increase performance at the possible expense of more lost aircraft. Another pilot-based cost that can be ignored in an unmanned fleet is the cost of changing plans, either while the vehicle is in air or on the ground. This should permit us to acquire more targets in a smaller amount of time, as aircraft plans can be changed to reflect new target information. One possible drawback that we might experience is that the loss of aircraft to a high-risk mission at the beginning of the campaign might decrease the total amount of target damage due to a lack of vehicles toward the end of the 7 days.

Scenario: Baseline of 313 targets and 78 aircraft, but with risk aversion turned off and the cost of replanning set to 0.

Results: The overall metric used to evaluate the performance of each trial was the aggregate target damage achieved over the course of the 7-day scenario. Costs relating to the attrition of aircraft and the utilization munitions was included in the evaluation as well. These metrics all correspond to Level-3 performance—with top-level military relevance to the overall air operations mission. As shown in Figure 8.4-13 and Table 8.4-8, the all-UCAV fleet managed to gain a significant increase in target damage when compared with the 313-target baseline. Another performance gain was in the amount of time needed to achieve a specified amount of target damage.

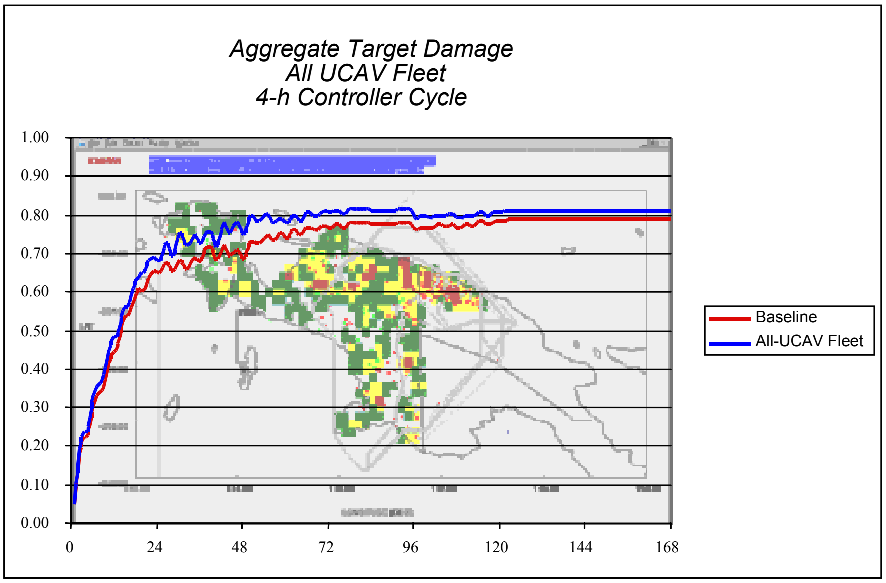


Figure 8.4-13. All-UCAV Fleet vs Baseline.

Table 8.4-8. 313-Target Scenario, Baseline vs All-UCAV Fleet.

Day Number	1	2	3	4	5	6	7
Baseline Scenario	0.665	0.703	0.780	0.782	0.808	0.813	0.813
All-UCAV Fleet	0.680	0.754	0.808	0.802	0.807	0.811	0.811

As is shown by comparing Figure 8.4-14 and Figure 8.4-15, there were many more aircraft mission changes for the all-UCAV fleet, both while the aircraft were on the ground and while they were in flight. Also evident is an increase in the total number of missions planned. Many missions that the UCAV fleet took on would have been discarded by the baseline controller due to high risk of aircraft loss.

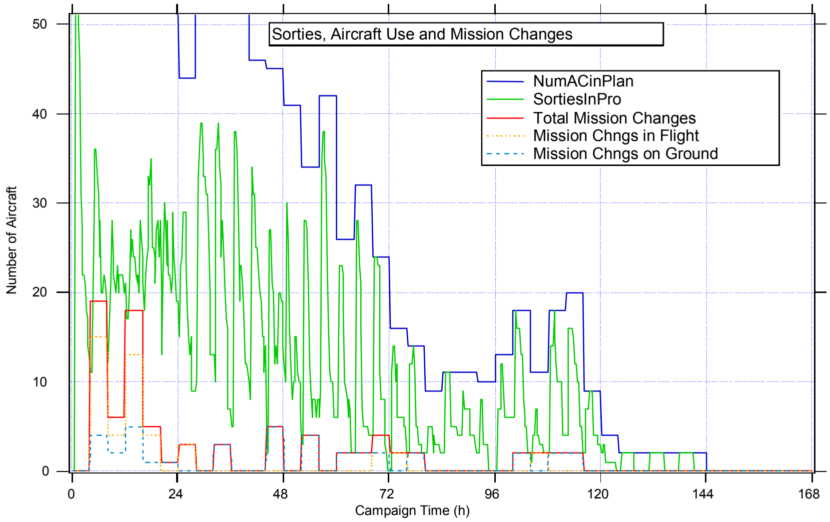


Figure 8.4-14. Baseline 4-h Cycle.

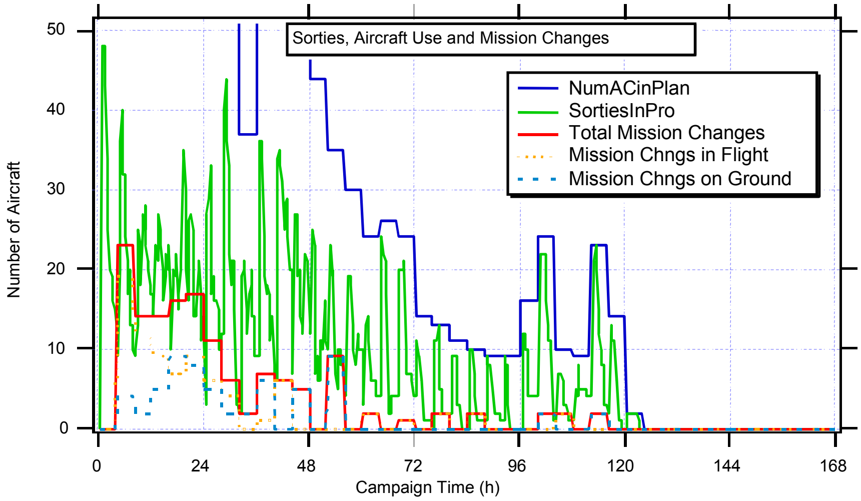


Figure 8.4-15. All-UCAV Fleet.

Conclusions: Removing the risk factor from the controller, in addition to the change in replanning cost, obviously resulted in a performance boost in target damage. The increase in the number of missions permitted by the controller can account for this change. Likewise, removing the cost of replanning permitted the controller to send already tasked vehicles to new, higher valued targets. In a high-risk situation where target damage must be maximized, an unmanned operation is an effective tool with which the objective can be achieved.

8.4.5.2 Time-Sensitive Target Sensitivity

8.4.5.2.1 All Targets-Time Sensitive

Hypothesis: A bound for the difference between the effectiveness of a 4-h cycle time and an 24-h cycle time can be generated by setting all targets to be time-sensitive. Closing the loop improves the controller's ability to respond to the time-sensitive targets.

Scenario: We executed two different types of trials: one at a 4-h loop closure rate and the other at a 24-h loop closure rate. Both used the standard baseline of 313 targets and 80 aircraft, only all targets have been changed to be time-sensitive.

Results: The overall metric used to evaluate the performance of each trial was the aggregate target damage achieved over the course of the 7-day scenario. Comparisons between the 4-h loop closure and 24-h loop closure show that almost immediately, the 4-h cycle starts to achieve a higher aggregate target damage (see Figure 8.4-16).

By the end of the 7-day campaign, the 4-h loop closure had achieved 45% more aggregate target damage than the 24-h closure rate. This extreme example shows the limit of improvement between the 24-h and the 4-h controller cycle. You can see the comparison with the standard base case, where approximately 10% of all targets are considered time sensitive, in Figure 8.4-16.

Conclusions: The controller operating at a 4-h closure rate was able to make more effective use of the time available for a time-critical target. Targets that showed up after the 24-h controller had planned were not accessed until the following planning cycle. Often, the target had expired by the time the controller was able to plan again. Therefore, many targets were never looked at by the 24-h controller. The 4-h controller had a similar problem, although on a much smaller scale. Some targets were not reachable at planning time because they would have already expired by the time a package was made ready and was able to reach the target. Obviously, a smaller loop closure rate permitted more time-critical targets to be accessed while they were still valid and reachable.

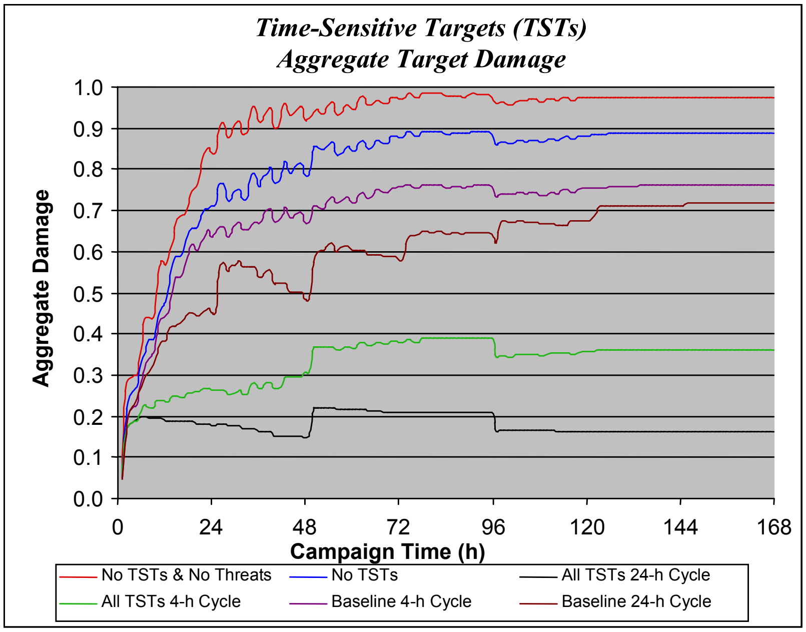


Figure 8.4-16. Time-Sensitive Targets.

8.4.5.2.2 No Time-Sensitive Targets

Hypothesis: The controller will prosecute all targets that are physically reachable by the aircraft. Some targets are not reachable due to the fuel constraints of the aircraft.

Scenario: We will run a baseline case of 313 targets and 78 aircraft where no targets are time-sensitive. Therefore, the controller need only plan paths around the existing threats in order to destroy a target.

Results: The overall metric used to evaluate the performance of each trial will be the aggregate target damage achieved over the course of the 7-day scenario. With no time-sensitive targets, the controller was able to send packages after all targets except those outside the aircraft's range. This range is mainly dependant on fuel constraints, which is likewise dependent on the current tanker locations. Some targets may have been ignored by the controller because no valid path around the threats is possible.

As you can see in Figure 8.4-16, the aggregate target damage achieved by this run is higher than all previous runs, although less than when no threats and no time-sensitive targets are present.

Conclusions: From this experiment, we learned some interesting things. First, we showed that there does exist some targets in our scenario that cannot be reached. Either they are physically out of range or they are surrounded by threats that cause the controller to decide that the cost of achieving them is too high.

8.4.5.2.3 No Threats and No Time Sensitive-Targets

Hypothesis: All targets will be prosecuted, except those that are unreachable, even with direct routing, giving us the highest possible amount of aggregate target damage.

Scenario: We ran a baseline case of 313 targets and 80 aircraft where no targets are time-sensitive and no threats are present. The controller need only plan direct route paths to the targets.

Results: In this scenario, all targets that are physically reachable by the aircraft were hit. Since no threats were present, all routes to the targets were direct and no aircraft were lost during the campaign. Therefore, the only reason a target would not be prosecuted is that the target was outside the strike range of the aircraft given the aircraft's fuel constraints with the current tanker locations. As it ends up, all targets in our Cyberland scenario were reachable. As seen in Figure 8.4-17 and Table 8.4-9, without targets disappearing and without threats, virtually all the targets are accessible to prosecution as shown in the upper curve. With the normal threat laydown and the routing for risk mitigation that is required, approximately 10% of the target set is not accessible to prosecution, even without time perishability, for the specified tanker locations, aircraft ranges, and weaponeering requirements. The middle curves show the nominal performance for 4-h vs 24-h loop cycles, with the former performing significantly better overall, and especially in the category of time-sensitive targets. The limit of this improvement is shown in the lowest curves, where a factor of two is seen for the difference between 4-h and 24-h cycles for the case of all time-sensitive targets.

Conclusions: The results illustrate an upper bound on the aggegate target damage that can be achieved.

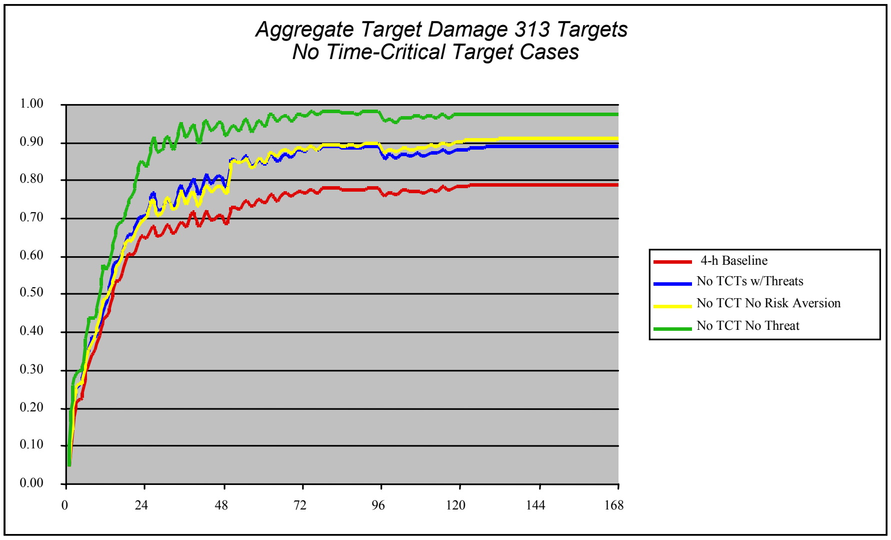


Figure 8.4-17. Maximum Feasible Aggregate Target Damage

Table 8.4-9. 313-Target Scenario, Baseline vs. All UCAV Fleet

Day Number	1	2	3	4	5	6	7
Baseline Scenario	0.665	0.703	0.780	0.782	0.808	0.813	0.813
No Time-Sensitive Targets, w/Threats	0.488	0.584	0.709	0.783	0.829	0.885	0.894
No Time-Sensitive Targets, No Risk Aversion	0.698	0.767	0.883	0.885	0.904	0.91	0.91
No Time-Sensitive Targets, No Threats	0.84	0.922	0.971	0.969	0.975	0.976	0.976

8.4.6 Improved Plan Generation

8.4.6.1 Negotiation

Hypothesis: Target and aircraft negotiation between the Level-1 and Level-2 controllers provides improved performance over a "one-pass" plan generation process. The discrepancy in plan quality between the centralized or undecomposed planner and the decomposed planner should be reduced.

Scenario: We employed target, aircraft, and mixed (aircraft and target) negotiation schemes between the Level-1 and Level-2 heuristic planners to the baseline scenario. Target negotiation reassigns a maximum of 5 targets per iteration, with a fixed set of 3 iterations per planning cycle. The pricing policy used to adjust the price on aircraft type per air base was to increase the price on each aircraft type by 20% if its use was greater than 70% and to decrease the price by 20% if it was used less than 30% of the time. Resource negotiation consists of a fixed set of 5 iterations per planning cycle. Five iterations of each type of negotiation were performed per planning cycle in experiments combining resource and target negotiation.

Results: The overall metric used to evaluate the performance of each negotiation trial was the aggregate target damage achieved over the course of the 7-day scenario. Figure 8.4-18 plots the aggregate target damage metric over the 7-day scenario for 3 sets of runs: the baseline set, the undecomposed set, and a target negotiation set (see also Table 8.4-10). Each set consists of 4 Monte Carlo trials. Negotiation closes the gap between the "single-pass" decomposed planner and the undecomposed planner and did not result in an increase in en route target reassignments or friendly attrition.

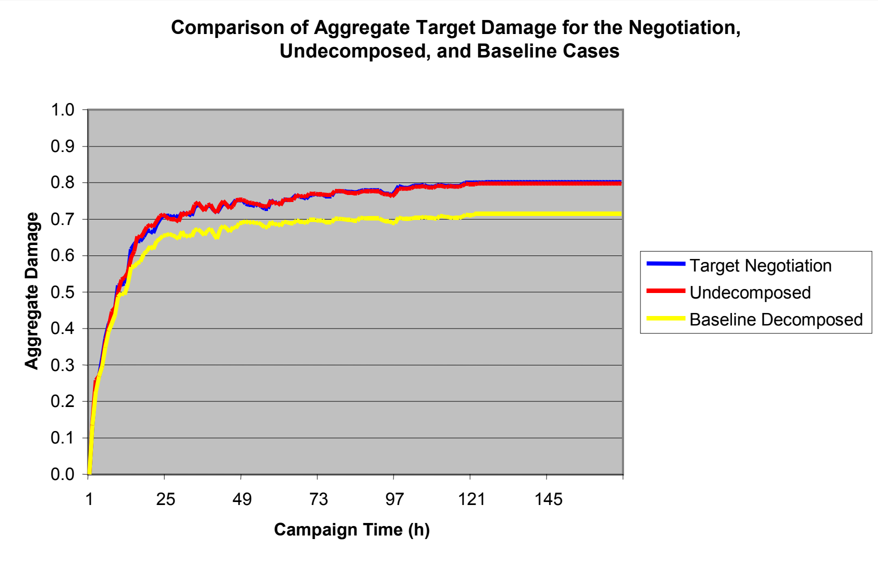


Figure 8.4-18. Comparison of Aggregate Target Damage of the Negotiation, Undecomposed, and Baseline Cases.

Table 8.4-10. 313-Target Scenario, Baseline vs All-UCAV Fleet.

Day Number	1	2	3	4	5	6	7
Baseline Scenario	0.665	0.703	0.780	0.782	0.808	0.813	0.813
Undecomposed Baseline	0.714	0.774	0.804	0.797	0.821	0.822	0.822
Target Negotiation	0.707	0.753	0.789	0.789	0.809	0.811	0.811

Computational performance of both the decomposition with negotiation and the undecomposed cases were comparable. The average time per run for these cases was 1.5 h as compared with 0.7 h for the baseline case.

We have very few resource and mixed negotiation results, as the runs were performed using an older version of the planner, which was computationally slow. Since we used a different version of the planner, it is not possible to draw direct comparisons to the runs presented above. We plot several cases: target negotiation, resource negotiation, alternating negotiation, sequenced negotiation, parallel negotiation, the baseline case, and the undecomposed case.

Alternating, sequenced and parallel negotiation refers to several different ways of combining target and resource negotiation. Alternating negotiation refers to target and resource negotiation taking turns one after the other to optimize the plan. Sequenced negotiation refers to having a single type of negotiation complete all its iterations before starting with the other type of negotiation. Parallel negotiation makes both target and resource swaps within a single iteration. Figure 8.4-19 and Table 8.4-11 show the results of each type of negotiation.

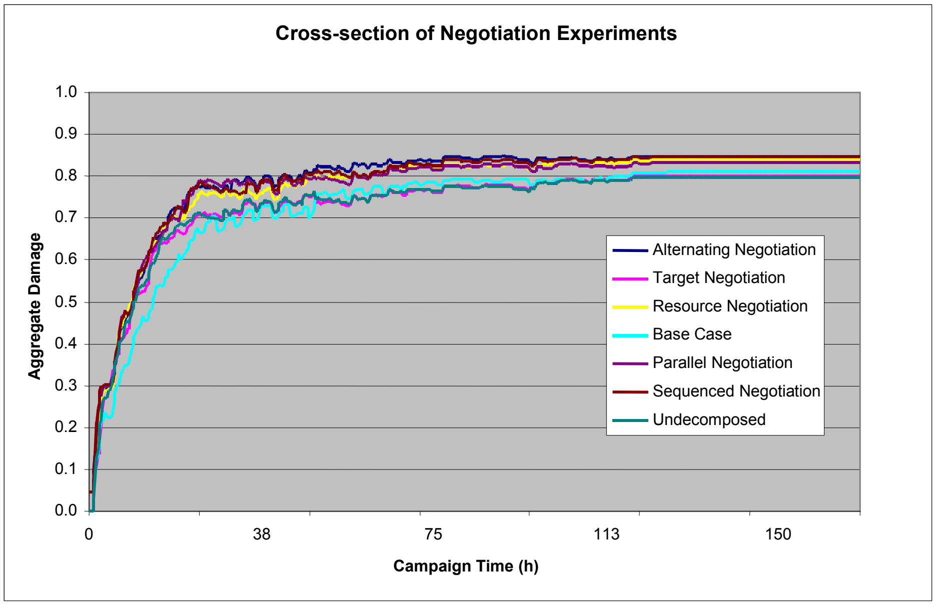


Figure 8.4-19. All Negotiation Experiments.

Table 8.4-11. All Negotiation Experiments.

Day Number	1	2	3	4	5	6	7
Baseline Scenario	0.665	0.703	0.780	0.782	0.808	0.813	0.813
Undecomposed Baseline	0.714	0.774	0.804	0.797	0.821	0.822	0.822
Target Negotiation	0.707	0.753	0.789	0.789	0.809	0.811	0.811
Resource Negotiation	0.757	0.795	0.825	0.817	0.838	0.839	0.839
Alternating Negotiation	0.776	0.808	0.841	0.836	0.846	0.848	0.848
Sequenced Negotiation	0.780	0.803	0.830	0.828	0.845	0.846	0.846
Parallel Negotiation	0.786	0.794	0.822	0.817	0.830	0.834	0.834

Resource and target negotiation perform equally well independently, but acted symbiotically when combined. We see that alternating negotiation and sequenced negotiation not only improves the baseline plan, but it is able to slightly improve on the results of the undecomposed planner as well. The main factor in this improvement appears to be the adjustment of resource prices, which encourages the sublevel planners to make better use of subproblem resources. Although the parallel negotiation scheme still improves on the baseline case, it performs the worst since it attempts to move both resource and targets at the same time. This allows for the possibility of the planner deciding to move resources into a subproblem to hit a set of targets, while at the same time, moving those targets to another subproblem.

Resource negotiation is based on the concept of swapping resources to the subproblem that can make the best use of it. Resource prices are initialized externally, but are updated per iteration of negotiation based on use. We maintain a resource price table for each type of aircraft in each subproblem. If a certain type of resource is being used extensively in a subproblem, its price is increased to encourage the subproblem to diversify its use of resources by using the less expensive aircraft. Aircraft are swapped among subproblems based on marginal benefit. We use the sensitivity information of each subproblem along with the current updated price of each resource type to determine how much more value will be obtained by the subproblem if an additional unit of a resource is brought in. We move resources from subproblems that value them the least to the ones that value them the most. We see that resource negotiation generalizes the concept of a subproblem, since aircraft stationed at an air base are not necessarily tied to a subproblem corresponding to the air base any longer. A subproblem can now consist of aircraft from multiple air bases. This adds one more dimension of freedom, since we are no longer constrained to optimize the problem by only shifting targets among subproblems.

Conclusions: We showed that good planning performance for a decomposed architecture is possible with negotiation. Notice that the curves tend to diverge toward the end of the first 24 h, when very few targets remain to attacked. We conjecture that negotiation is used most effectively when targets are sparse. The master-level target allocation algorithm performs its tasks in the same manner regardless of whether there are many or few targets. It may be possible for it to inefficiently assign a majority of targets to a single air base. We believe that negotiation "corrects" these inefficiencies by redistributing the targets to make the best use of all the resources. This was demonstrated in a test scenario with two types of targets and two air bases. The targets were lined up evenly with each air base capping an end of the target line. One type of target lined up on the right side was best suited for the air base on the right side, and the other type of target lined up on the left side was best suited for the air base on the left side. The master planner was initially instructed to flip-flop the targets so that each air base was assigned target on the "wrong" side. After several iteration of negotiation, the situation was corrected and the targets were each assigned to the appropriate base.

The performance of negotiation is hindered because of its greedy nature. Negotiation results in substantial improvements within a single planning cycle, however, these results are not compounded over time. Given the exact same starting state, planning with negotiation can result in a 10% improvement over a baseline plan. However, since the negotiated plan is optimized for just the current planning cycle, the subsequent planning cycles may suffer from lack of resources as a result. Figure 8.4-20 shows the improvement of a plan per iteration of negotiation of a single planning cycle.

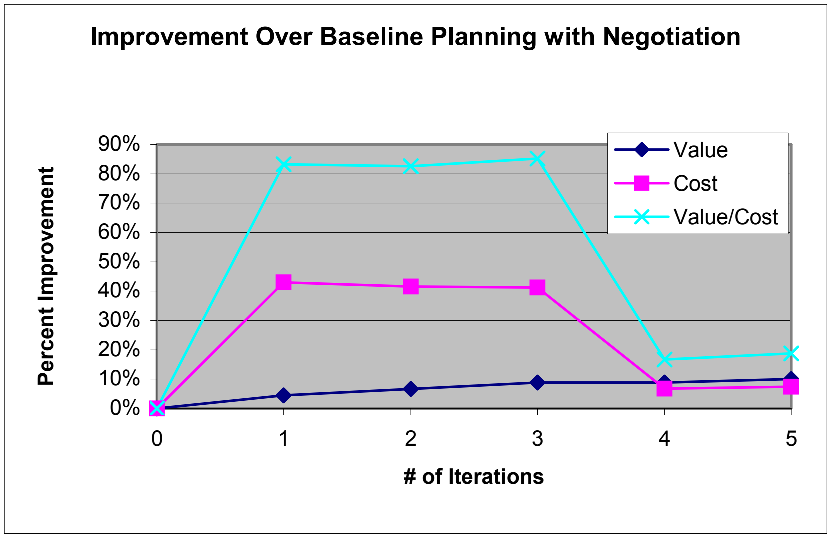


Figure 8.4-20. Improvement of a Plan per Iteration of Negotiations.

We see that for five iterations of negotiation, the value of the baseline plan is improved by about 10% and the cost is reduced by 40%. The benefits of negotiation are best illustrated with the value over cost curve, which shows a significant improvement in the efficiency of resource use. The sharp decline of improvement on the fourth iteration may have been the result of overshooting the optimal solution, since we used a fixed step size throughout the iterations. Decreasing the step size per iteration may lead to more consistent results.

Further experimentation is needed to improve our understanding of the dynamics of resource negotiation. Experimental results suggest that alternating negotiation and sequenced negotiation are most promising. As with target negotiation, a theoretical formulation of resource negotiation will be a useful part of developing a deeper appreciation of its intricacies.

8.4.6.2 Optimization-Based Planner

Hypothesis: Optimization-based planning improves performance compared with the heuristic Level-1 and Level-2 planners.

Scenario: Employ optimization-based planning algorithm and the heuristic for a 910-target scenario.

Results: The optimization-based planning algorithm generates more effective plans than the heuristic. The optimization plans achieve more target value and/or lose less aircraft.

As shown in Figure 8.4-21, the optimization-based planner achieves higher target value earlier in the scenario. The heuristic is able to catch up in the end, but this is because the number of targets available to hit is low, and resources are no longer scarce. At 72 h, the optimal planner has achieved 10% more target value than the heuristic.

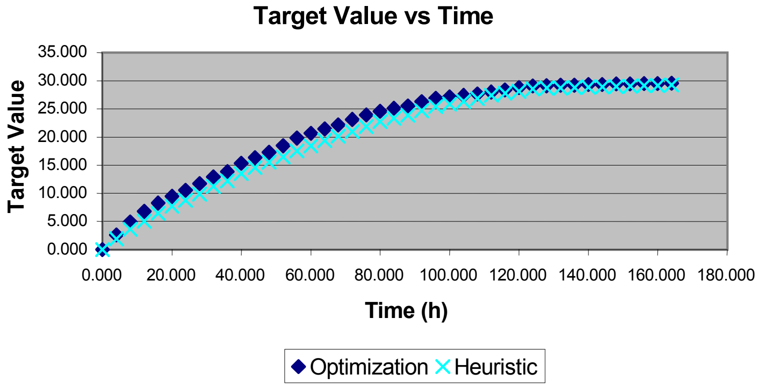


Figure 8.4-21. Target Value vs Time.

For an average planning cycle, the optimal planner took 4.8 min to generate a plan, while the undecomposed heuristic took 2.3 minutes.

Conclusions: We conclude that the optimal planner quickly generates effective plans as validated by comparing its performance with the heuristic. Also, the optimal planner method with composite variables can quickly generate plans for large scenarios.

8.5 Event-Based Loop Closure Experiments

The event-based loop closure experiments show good, sometimes remarkable, improvement in both accumulated target damage and aircraft attrition over that of the 4-h cyclic case. The target damage improvement comes mostly from the ability to prosecute more time-critical targets. The improvement in attrition comes mostly from the ability to adjust package configurations more frequently, which reduces risk, and issue mission abort commands sooner (before additional aircraft are lost). Although much of the experimentation has been carried out using a baseline case that exhibits fairly nominal behavior, some off-nominal experimentation (high-threat, unmanned aircraft) shows that the controller's performance holds up well. Also, a hierarchical, event-based controller was shown to exhibit the desired behavior of allowing each node in the hierarchy a certain amount of latitude in the control over its domain. The affect was that of isolating the high activity of lower levels, due to numerous smaller disturbances, from that of the higher levels.

8.5.1 Comparison to Cyclic Loop Closure Baseline

Hypothesis: Air operations campaign effectiveness is improved by addressing events (e.g., new time-sensitive targets, aircraft attrition, and missed targets) as they occur, rather than at a fixed replanning interval (e.g., 2 or 4 h).

Scenario: Baseline (313 targets, 80 aircraft) where the event-based controller architecture was employed, but with no negotiation between levels in the hierarchy.

Results: An overall comparison between the event-based and cyclic techniques is shown in Figure 8.5-1, where an improvement of roughly 15% is realized for the event-based case over that of cyclic. These results are broken down for each target group in Figure 8.5-2 and summarized in Table 8.5-1.

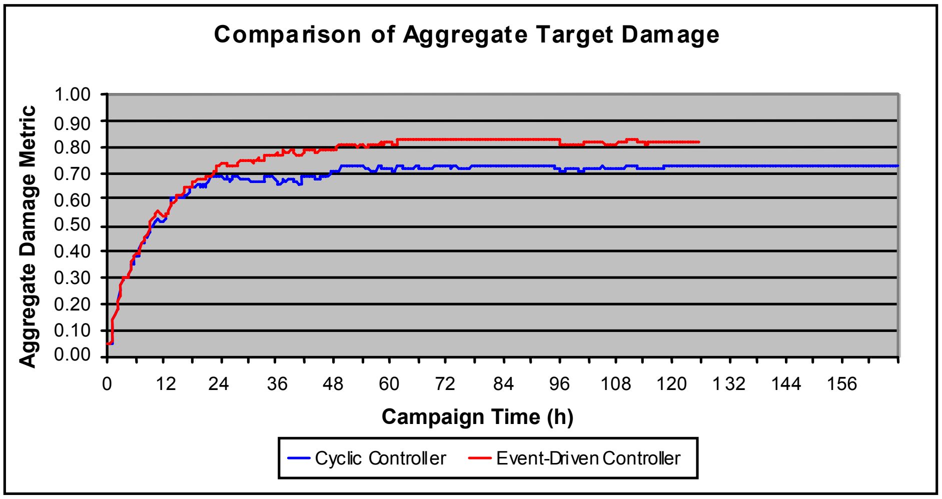


Figure 8.5-1. Aggregate Target Damage Compared for Event-Driven vs Cyclic Controllers.

Conclusions: As expected, when changes in the environment can be considered and addressed sooner rather than later, performance can improve. In this case, the gains come predominantly from the ability to prosecute more time-sensitive targets in a timely manner.

Hypothesis: Event-based replanning is less disruptive because it allows problems and opportunities to be addressed locally rather than globally as per the cyclic replanning approach.

Scenario: Baseline (313 targets, 80 aircraft).

Results: Planning events for our two-level controller are shown in Figure 8.5-3, with Level-1 results at the top of the figure and Level-2 results at the bottom. The baseline scenario has an embodied set of random events, some minor, others not, that dictate exactly how much and what kind of activity occurs at each level. Shown clearly is that each level in the controller does indeed monitor and act on its own set of disturbance events. Some fraction of the bottom level's events do precipitate replanning activity at the upper level, as it is "bubbled -up." In this case, the number of bubble-up events were relatively few.

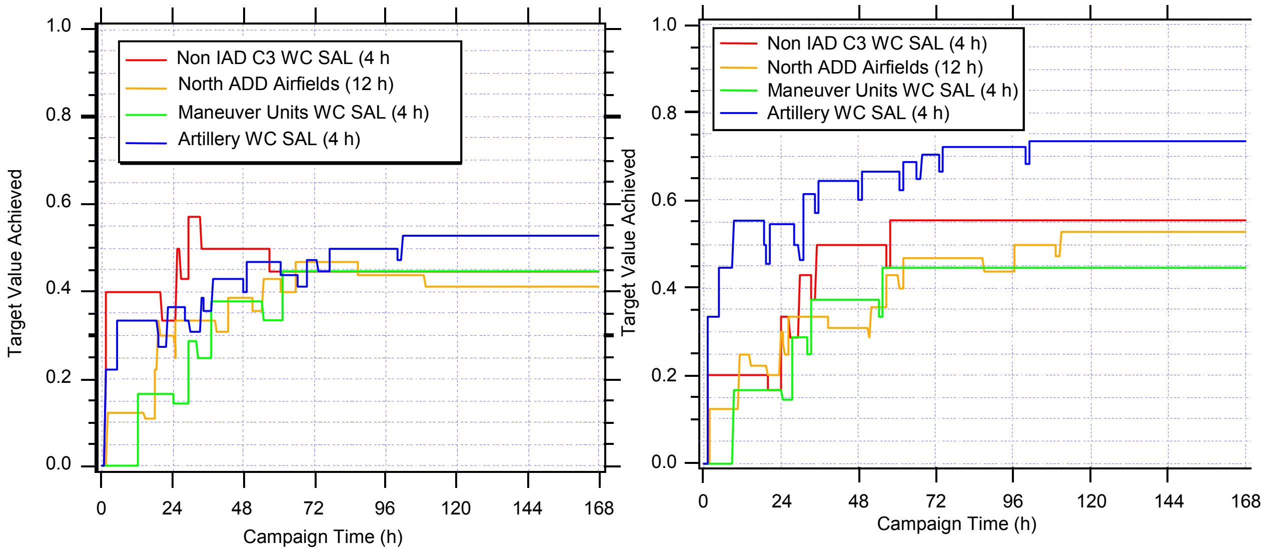


Figure 8.5-2. Categories of Target Damage Compared for Event-Driven vs Cyclic Controllers.

Table 8.5-1. Cyclic vs Event-Based Target Value Achieved.

Time Critical Target Category	4-h Cycle	Event-Based
Non-IAD Invasion Salient (t = 4-h)	0.45	0.56
North Air Defense District Airfields (t =12 h)	0.41	0.53
Maneuver Units Invasion Salient (t = 4-h)	0.45	0.45
Artillery Invasion Salient (t = 4-h)	0.52	0.74

Conclusions: As designed, the controller hierarchy addresses system disturbances at the appropriate level, but will refer problems it cannot solve to the next higher level.

8.5.2 Scenario Variations

All the results in previous sections used the standard baseline scenario, which was designed to be as realistic as possible, and with fairly nominal behavior. We conducted experiments with two scenarios that depart from this nominal regime, in order to explore the controller's robustness to off-nominal behavior. The first is a "high threat" case, where the threat level has been increased by a factor of five. The second removes from consideration all risk of losing aircraft and pilots by using an "all-UCAV" fleet of aircraft. In both these cases, with limited testing, controller performance appears to hold up. The details follow.

8.5.2.1 High Threat

Hypothesis: Event-based replanning combined with negotiation can provide significant improvement in performance over that provided by the 4-h cyclic controller without negotiation. In particular, event-based replanning should allow for package reconfiguration or mission aborts when a critical package escort is lost, and negotiation should allow targets to be reassigned to those subproblems that can address them at the least (attrition, fuel etc.) cost.

Scenario: Baseline (313 targets, 80 aircraft) but with threat lethality increased by a factor of 5.

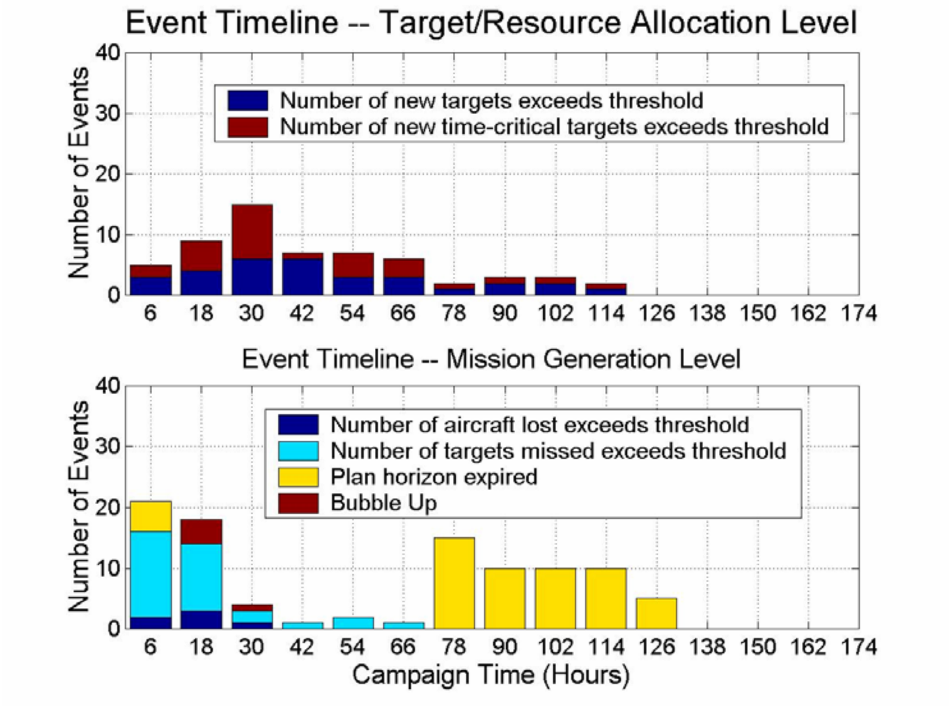


Figure 8.5-3. Aggregate Target Damage Compared for Event-Driven vs Cyclic Controllers

Results: The event-based controller achieves only slightly more target damage, but at significantly lower attrition cost, as shown in Figure 8.5-4 and Table 8.5-2. We might have expected to see a separation between the cyclic and event-based case more in line with the results of the previous section. It appears that the event-based approach lost proportionately more than the cyclic case did, perhaps due to the increase in threat, but it is not clear from the experiments we were able to perform. Additional testing is needed to better answer this question.

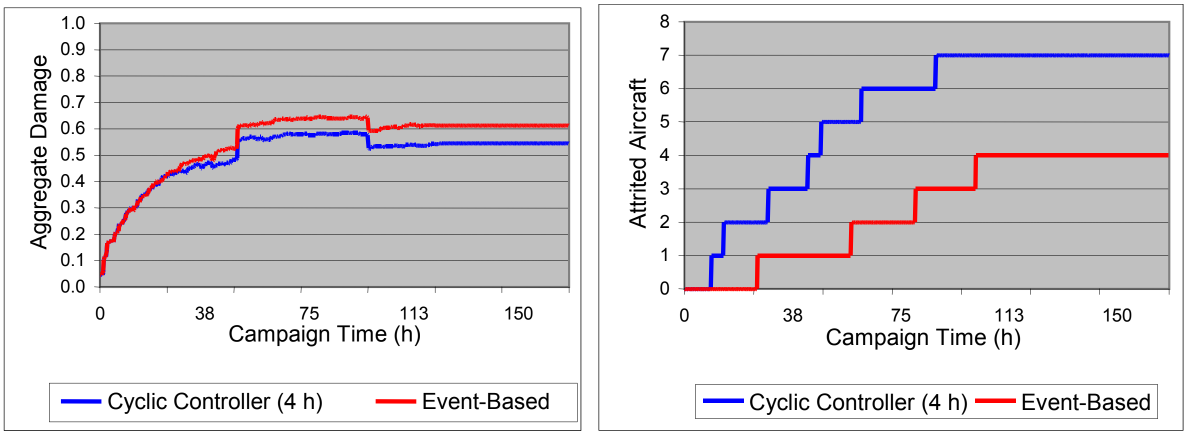


Figure 8.5-4. Comparison of Cyclic and Event-Based Controller Aggregate Damage and Aircraft Attrition in High-Threat Scenario.

Table 8.5-2. Comparison of Cyclic and Event-Based Controller in High-Threat Scenario.

Day Number	1	2	3	4	5	6	7
Cyclic	0.421	0.482	0.580	0.576	0.542	0.546	0.546
Event-Based	0.421	0.525	0.640	0.643	0.613	0.613	0.613

Conclusions: It does appear that good savings in total aircraft can be realized in the event-based case, where the controller has timely information on when individual aircraft are lost. Missions are aborted when it is known that key escorting aircraft are lost. The 4-h cyclic controller will unwittingly allow missions to go forward for a much longer time than the event-based controller will, with an increased likelihood that one or more aircraft in the package will be shot down. This, of course, raises the attrition rate and slightly lowers the aggregate target damage. Evidently, the aircraft that are conserved with the event-based controller enable this higher yield. This experiment produces the contributions of a competent squadron leader who would not exceed the allowable risks of the mission anyway.

8.5.2.2 All-UCAV Fleet

Hypothesis: When missions are unmanned, as is the case with an all-UCAV fleet, the reluctance to subject an aircraft and pilot to high risk is removed, freeing the controller to trade increased attrition for increased performance. Here, the two important aspects of performance are total target damage (the more the better) and the time frame over which the damage is inflicted (the shorter the better).

It is not clear whether lifting the constraint on risk will precipitate an increase in target yield or not. This is because of the greedy nature of the planning algorithm. In its quest to maximum near-term target damage, it could lose enough aircraft to seriously jeopardize its ability to address the stream of future targets. However, we suspect that performance gains are likely because of the relatively conservative levels of acceptable risk that have been set in most of the experiments.

Scenario: Baseline (313 targets, 80 aircraft) but with package risk aversion turned off in the controller.

Results: Plots comparing target damage between the baseline scenario and an all-UCAV fleet, for both cyclic and event-based controllers, are shown in Figure 8.5-5 and summarized in Table 8.5-3. Evident is a significant increase in target damage and a remarkable reduction in the time to levy this damage. This performance boost evidently comes from a 55% jump in the number of missions mounted over that allowed in the event-based, baseline case. When missions are manned, the risk averse strategy prevents many missions from continuing when aircraft are lost en route or even being planned in the first place.

Conclusions: Removing risk from consideration has raised the number of missions to 581 (from 376), at a cost of raising the number of lost aircraft to 10 (from 1). It is the increase in the number of missions allowed that no doubt provides the overall boost to performance. Although the increase in lost aircraft is tenfold, it amounts to just around 12% of the fleet. This loss rate is usually unacceptable for manned missions, but can be considered reasonable for unmanned missions.

It appears that if manned operations force a very risk-averse strategy, then maximizing the number of unmanned operations will have significantly more impact and should probably be pursued. The only caveat is if there are pilot contributions to target damage that have not been modeled in these studies. That is, are manned aircraft somehow more effective at executing missions once they are planned?

8.5.3 Improved Plan Generation

Experiments to verify that the negotiation capability and the optimization-based planner still perform at least as well as they did for the cyclic case, if not better, have been performed. They are described in the following two sections.

8.5.3.1 Negotiation

Hypothesis: Combining negotiation with event-based replanning provides improvements beyond those provided by either one alone.

Scenario: Baseline (313 targets, 80 aircraft). Here we employ the interlevel negotiation capability in the event-based controller architecture.

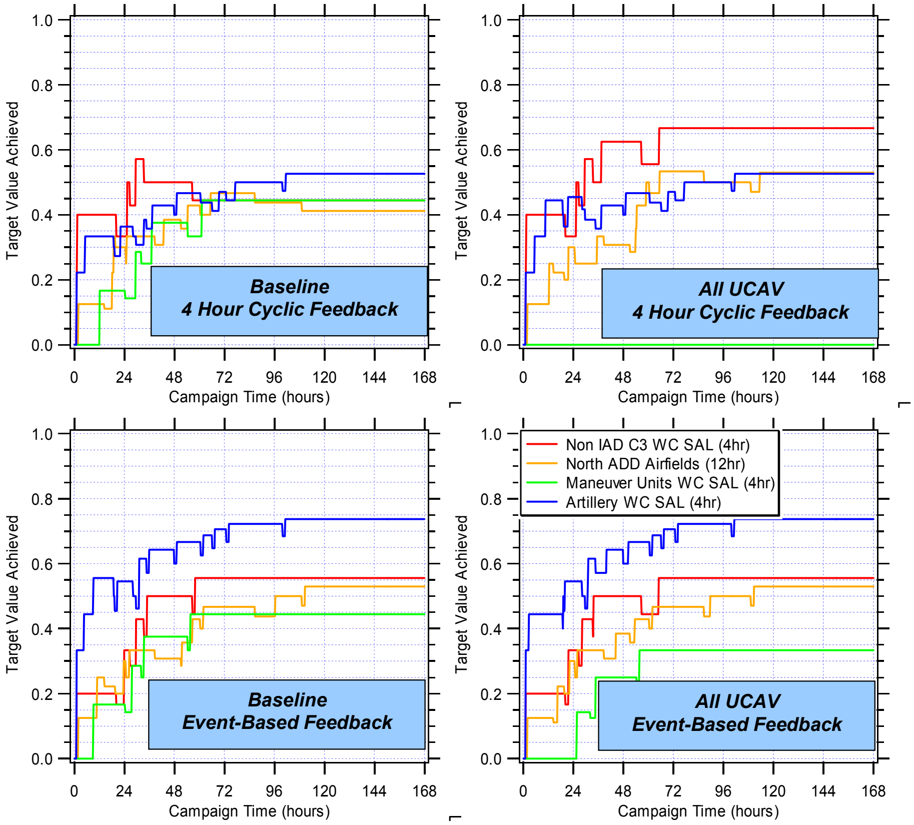


Figure 8.5-5. Planner Decomposition Specification.

Table 8.5-3. Planner Decomposition Specification.

Experiment	Total Missions	Target Damage	Aircraft Attrition	Time to 0.5 Damage
Baseline 24 h	266	0.39	2	~
4-h Cycle	344	0.50	5	2 days
Event-Based	376	0.53	1	1.5 days
All-UCAV	581	0.72	10	10 h

Results: The results of the single trial we ran using event-based replanning with negotiation are encouraging. Figure 8.5-6 and Table 8.5-4 compare the aggregate target damage for three cases: (1) event-based replanning without negotiation, (2) event-based replanning with negotiation, and (3) cyclic replanning with negotiation. We see that event-based replanning with negotiation results in a slight improvement in plan value over using non-negotiated event-based replanning. The main advantage of the combination of event-based replanning with negotiation appears in the time period spanning from 24 h to 96 h. The aggregate target damage achieved with the cyclic controller lags behind significantly since it only gets updates of new targets every 4 h. We see that the cyclic planner eventually catches up once it learns about the remaining targets. In addition, Figure 8.5-7 shows the attrition profile over the course of the scenario. The attrition as a result of the event-based controller is significantly less than that of the cyclic controller.

Conclusions: These results suggest that event-based replanning combined with negotiation improves on the performance provided by either one alone. Event-based replanning makes the planner more responsive to changes in the state, and negotiation provides further enhancement to the baseline solutions of the event-based planner.

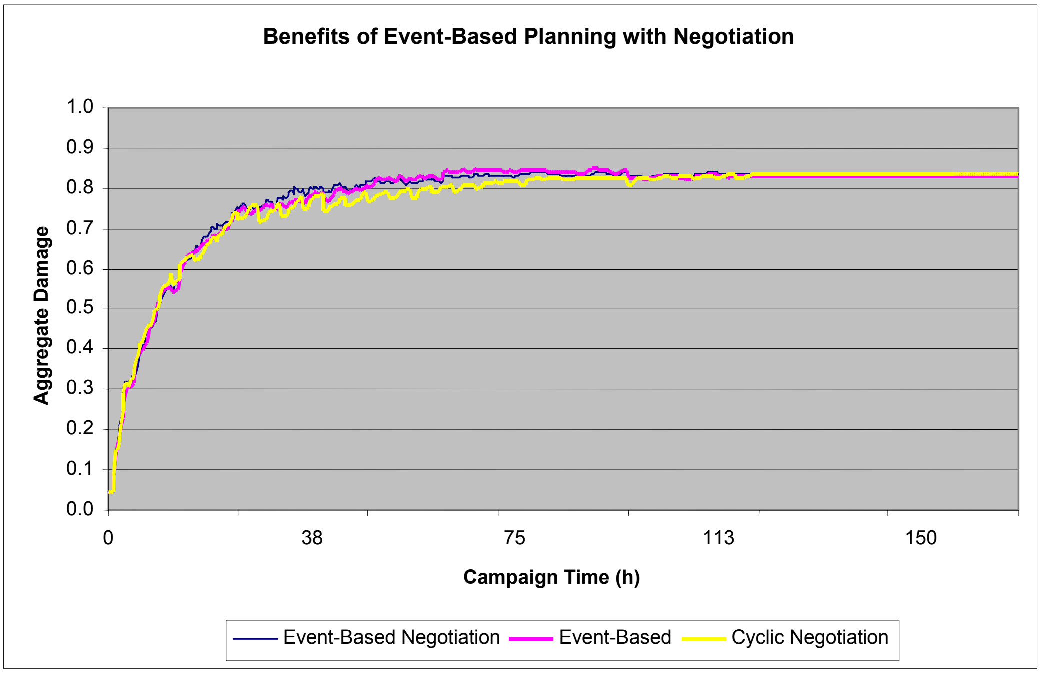


Figure 8.5-6. The Affect on Aggregate Target Damage from Interlevel Negotiation.

Table 8.5-4. Comparison of the Effect of Interlevel Negotiation.

Day Number	1	2	3	4	5	6	7
Event-Based w/Negotiation	0.752	0.816	0.833	0.837	0.836	0.836	0.836
Cyclic w/Negotiation	0.723	0.768	0.813	0.813	0.834	0.835	0.835
Event-Based	0.743	0.804	0.844	0.844	0.831	0.831	0.831

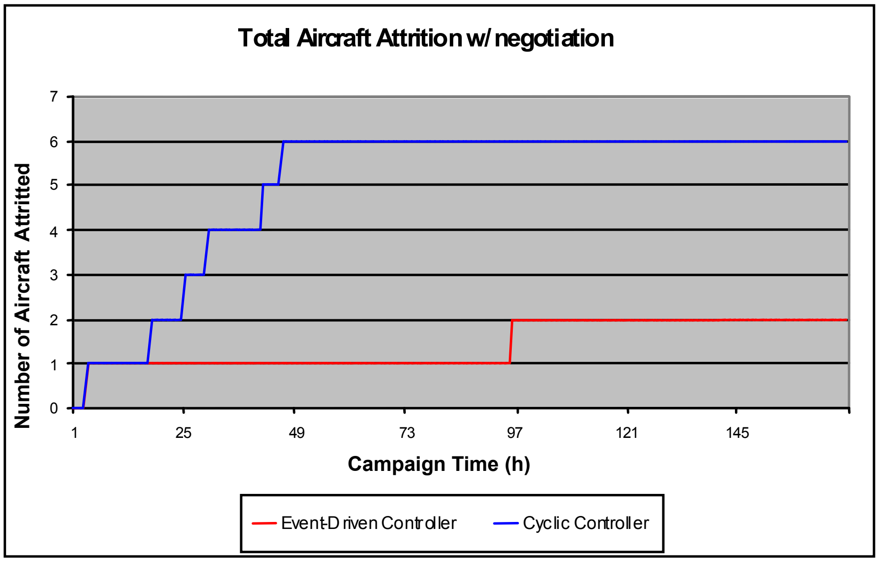


Figure 8.5-7. The Affect on Aircraft Attrition from Interlevel Negotiation.

It appears that negotiation is most beneficial when there is a relatively small set of targets available. We see that event-based replanning with and without negotiation seem to track one another throughout the scenario, except at the very end of the campaign. We expect that, since fewer targets remain toward the end of the scenario, negotiation is able to make better target assignments and improve the baseline plan of the event-based planner. The combination of negotiation with the event-driven controller also results in lower attrition. Although it is not known for certain, this phenomenon could be attributed to the ability of the event-driven controller to react to new targets that are in less threat intense regions. By replanning to hit the "safer" targets when they appear, the event-based controller may be able to achieve the same aggregate damage as in the cyclic controller with a lower cost. Further experimentation would help to characterize situations where this combination of event-based planning and negotiation offers the greatest advantage.

8.5.3.2 Optimization-Based Planner

Hypothesis: Employing the optimization-based planner within the event-based controller architecture should result in improved performance over that achieved by the optimization-based planner in the 4-h cyclic controller. Again, performance improvements are expected by the controller's improved ability to address new time-sensitive targets and attrition events in a timely manner.

Scenario: Baseline (313 targets, 80 aircraft).

Results: Aggregate target value and number of aircraft lost are plotted in Figures 8.5-8 and 8.5-9, and are consistent with the results obtained for the heuristic controller (see also Table 8.5-5). The optimal algorithm may have performed a little better for the cyclic case, but the event-based performance is essentially the same. The more significant improvement is, again, in the savings in aircraft. Three aircraft have been conserved as a result of going to an event-based approach.

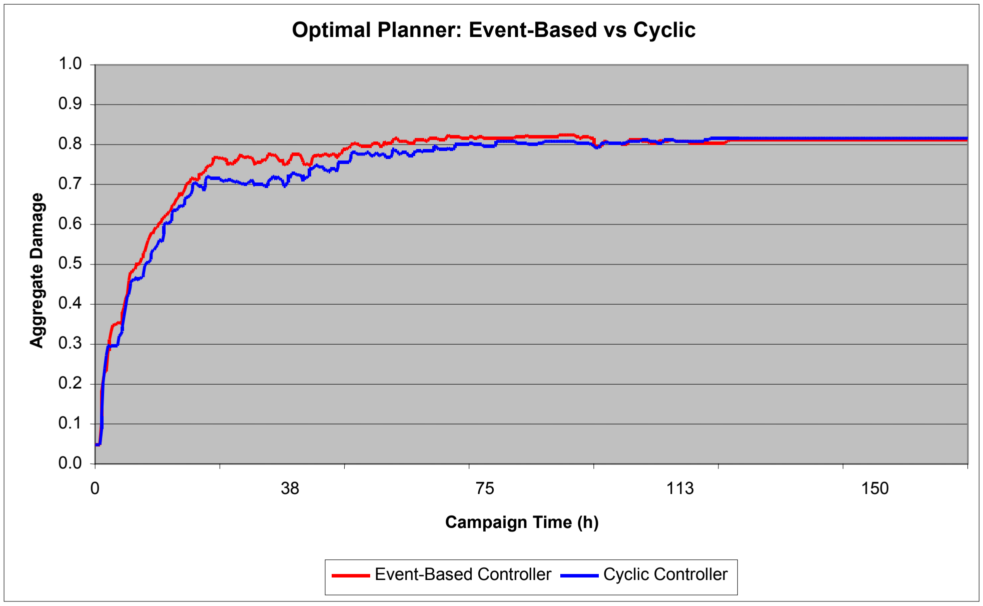


Figure 8.5-8. Comparison of Target Damage for the Optimization-Based Planner (Event vs Cyclic).

Table 8.5-5. Comparison of the Cyclic vs. Event-Based Planning Approaches with an Optimization-Based Planner

Day Number	1	2	3	4	5	6	7
Event-Based	0.765	0.788	0.818	0.814	0.805	0.814	0.814
Cyclic	0.714	0.754	0.800	0.797	0.815	0.816	0.816

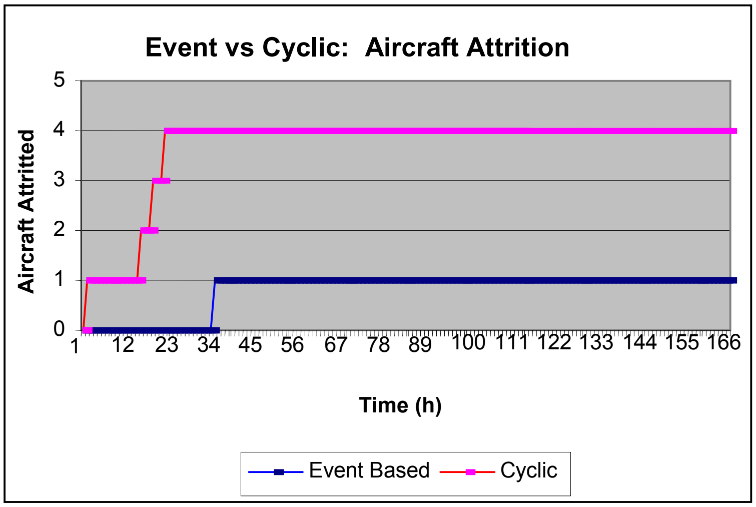


Figure 8.5-9. Comparison of Aircraft Attrition for the Optimization-Based Planner - Event vs Cyclic.

Conclusions: In addition to supporting our hypothesis that event-based planning outperforms the cyclic approach, it also verifies the correct operation of our optimization-based planner within the event-based controller architecture. For problems where resource allocation is not handled adequately using a heuristic approach, the optimal approach can now be employed. The heuristic approach has its own advantages for certain planning situations and would be used in those cases.

8.6 Air Operations Enterprise Model (AOEM) Calibration Experiments

The AOEM calibration experiments were two sets of scenarios that were constructed to be executed using the Draper JFACC Air Operations Controller with the AOEM simulation for comparison with execution of the controller with the Draper JFACC Air Operations simulation. The purposes of the activity were:

- To understand AOEM/Draper model mismatches, specifically in the areas of threat interactions target damage.
- To ensure correct interfacing of the AOEM to the Draper controller.
- To increase confidence in modeling and model implementation by comparison between simulations.
- To examine the performance of the controller in different simulation environments prior to undertaking AOEM parametric experiments.

All the calibration experiments used the heuristic cyclic planner. Two scenarios were used:

- Simple geometry calibration experiment.
- Modified Cyberland scenario experiment.

The objectives of the simple geometry experiment were:

- To determine effects of detailed AOEM model (surveillance radars, revisit intervals, fire control radars, launchers, subsector commanders, and fire control logic, etc.).
- To determine state file parameters to send to the controller to achieve the same aggregate threat engagement result.

The objectives of the Modified Cyberland scenario experiment were:

- To compare aggregate results of AOEM and Draper simulations with maneuver units held static in AOEM and weaponeering adjusted to minimize differences in target damage models.

- To add in AOEM features of maneuver unit movement, attrition of air defenses, attrition of IAD C^3 to examine the ability of the controller to implement automated taskings for different campaign objectives and phasings and to look at effects of decision cycle time.

The experiments were extremely useful in tracking down a number of interfacing as well as modeling discrepancies, and greatly increased the confidence in using the Draper controller with the AOEM. A number of corrections and improvements to the AOEM models also resulted from this exercise.

8.6.1 Simple Geometry Scenario

The AOEM uses a discrete model of threat location and an elaborate description of commander logic with subsector organization and fire control logic, etc. The Cyberland scenario was much too complex a scenario in which to apply the AOEM, so a simple scenario was constructed with the following features:

- No refueling, single base, single target location, no risk mitigation.
- 100-km thick SAM belt: 127 SAM systems on hexagonal grid.
- Parametric variations, including uniform packages of various compositions.
  - No escorts, strikers only.
  - Strikers and HARM-shooting weasels only.
  - Strikers, HARM-shooting weasels and jammers.
- Fixed SAM parameters only or mobile SAM parameters only.

Figure 8.6-1 illustrates the air defense laydown that was defined for the AOEM. Note that the target is at the center. The corresponding threat density representation that was derived for use in the Draper simulation as well as the controller is depicted in Figure 8.6-2.

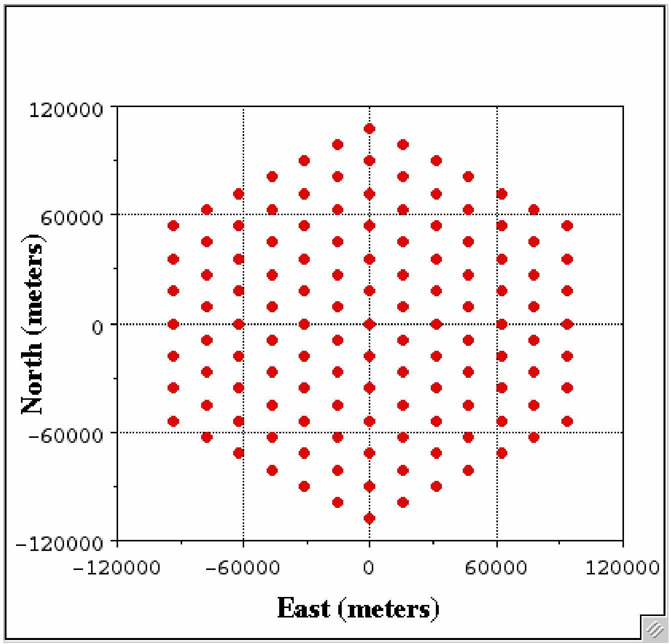


Figure 8.6-1. Discrete SAM Location Map for AOEM.

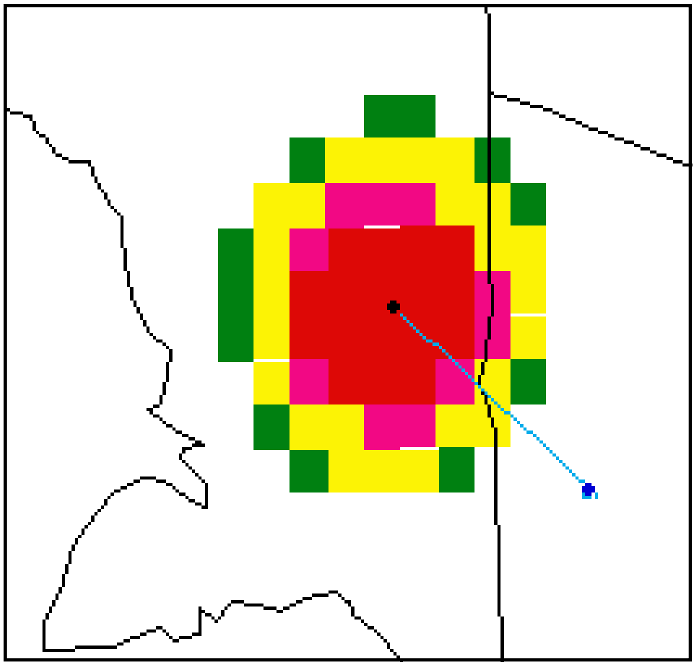


Figure 8.6-2. Probabilistic Threat Density on 30-km Grid.

The target list included 300 targets, all with location specified as being at the center of the SAM distribution. In effect, the set of missions represented an implicit Monte Carlo since the same mission would be repeated multiple times and the statistical outcome analyzed to reveal model features and model discrepancies. Each experiment would run until attrition precluded the creation of the specified packages for that experiment.

A set of parametric studies was done with different scaling of the threat parameters. Because of "saturation" effects, relating to the circumstance of many SAM sites simultaneously engaging a single aircraft, one of the more meaningful results was the scale factor =30 results. The "scale factor" is a parameter that divides the probability of kill per SAM engagement. With a scale factor =30, the P_K for fixed SAM sites was 0.01. Significant differences between parameters for fixed vs mobile SAMs was that the range of the former gave rise to a high degree of overlapping coverage, and AOEM results were particularly sensitive to Commander logic that apportioned decisions between sector Commanders and subsector Commanders.

The results of the simple geometry experiments are summarized in Table 8.6-1.

Table 8.6-1. Comparison of Results for Simple Geometry Calibration Experiments.

Description	Engagement per Mission		Loss Rate per Mission		Risk per Mission	
	AOEM	Draper	AOEM	Draper	AOEM	Draper
Fixed SAM Threats Striker Only	44.3	47.6	0.202	0.256	--	0.24
Mobile SAM Threats Striker Only	23.9	23.8	0.202	0.215	--	0.16
Fixed SAM Threats Striker+HARM	29.5	47.8	0.170	0.272	--	0.19
Mobile SAM Threats Striker+HARM	10.6	23.9	0.068	0.190	--	0.13
Fixed SAM Threats Striker+HARM+Jammer	1.6	2.4	0.004	0.011	--	0.0096
Mobile SAM Threats Striker+HARM+Jammer	1.8	2.4	0.012	0.012	--	0.013

The first set of numbers in each column are the AOEM results; the second are the Draper results.

There is reasonable correspondence between Draper models and AOEM models despite substantial differences in modeling approaches! The most significant discrepancy is that the Draper threat attrition model handles the effects of weasel-only escorts more conservatively than the AOEM. In other words, the Draper model predicts lower effectiveness leading to more engagements and losses if packages with only weasel escorts are used. The seeming discrepancy for the full-escort cases 15 and 18 are understandable in view of rather poor attrition statistics for the AOEM (e.g., only 4 losses).

The significance of these results relates to the uses of the threat model in the Draper controller:

- To find minimum risk fuel-feasible routings with necessary tanker stops.
- To yield mission timeline, absolute risk, tanker requirements.
- To choose missions to maximize value/cost that include attrition.
- To eliminate missions that exceed estimated absolute mission risk.

The differences in SAM engagement modeling in AOEM and Draper models (e.g., sector command, handoff, firing doctrine) most likely preclude the universal calibration of all metrics for all scenarios. On the other hand, attrition loss rate/mission can be calibrated *reasonably* well.

8.6.2 Modified Cyberland Calibration Experiments

The Cyberland scenario was modified to include weaponeering derived from the AOEM models, the SAM air defense threat models calibrated from the simple geometry experiments, and a set of 31 package specifications derived from the parameters used by the AOEM. There were also some modifications to the Commanders Intent weightings. Overall, the targets in the modified Cyberland scenario required considerably more weapons on target, and the packages were constructed to provide options to meet these requirements.

The screen snap of the mission tracks is seen in Figure 8.6-3 to pretty much resemble the display produced for the original Cyberland scenario in the Draper JFACC Air Operations Simulation.

The aggregated results comparing the Draper simulation results on this scenario with those from the 31 January Bolt, Beranek & Newman (BB&N) AOEM archive for threat scale factor = 30 are shown in Table 8.6-2. The AOEM sorties were estimated since the reporting on the archival report was fragmentary. It can be seen that the results are reasonably close despite the large differences in modeling approaches between the two simulations.

8.6.3 Calibration Experiment Conclusions

The conclusions from this exercise were:

- Diligent efforts required to identify and resolve modeling and interface issues resulted in:
    - Consistent results for the AOEM and Draper models.
    - Confidence that the AOEM/Draper controller can yield plausible results as campaign objectives and constraints are varied.
- The process to identify and resolve modeling and interface issues took *significantly* more time and resources than originally anticipated.

Definition of joint experiments exercising both simulations on common experiments was critical to revealing undocumented modeling assumptions and greatly assisted in bridging the *semantic gap* between the two models.

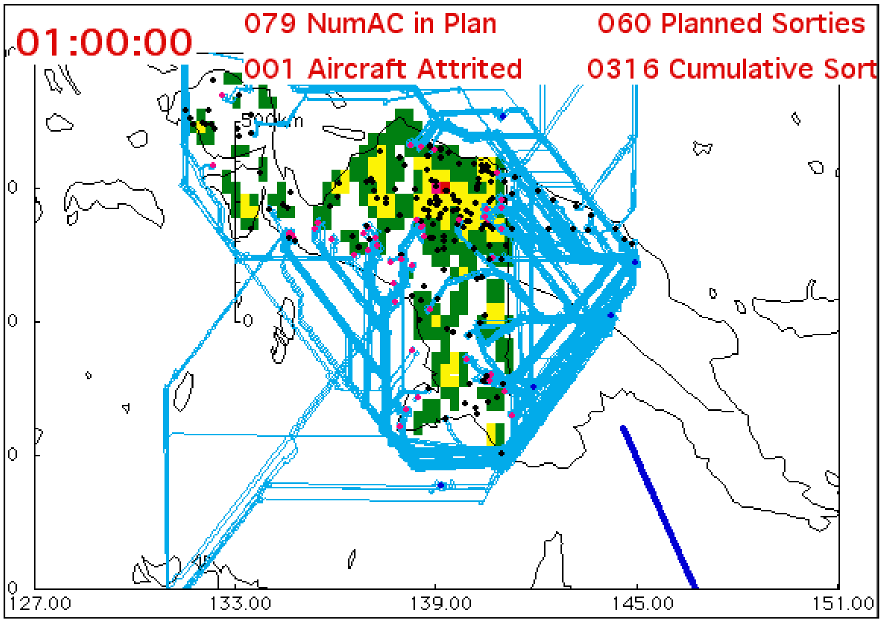


Figure 8.6-3. Modified Cyberland Scenario Calibration Experiment.

Table 8.6-2. Comparison of Results for Modified Cyberland Scenario Calibration Experiment.

Result	AOEM	Draper
Total Sorties Flown (119 h)	1104	1008
Total Aircraft Shot Down	0	1
Total Number Targets Hit	235	214
Aggregate Target Damage	0.80	0.829
Number MK-1+MK-2s Used	3503	3893

9 Conclusions

There has been a long-standing advocacy by those in the command and control community for closing the command and control feedback loop at ever shorter intervals. This advocacy relates to a *rule of thumb* in control system design: that is, to achieve good system performance and maintain robustness, the total time lag for each control cycle attributable to: (1) measuring and conditioning feedback signals, (2) information transfer, and (3) control law computation should be one-fifth to one-tenth the time constant of the fastest dominant mode of the plant to be controlled. Applying that rule of thumb to the control of military air operations, our goal should be command and control cycle times that are five to ten times shorter than those of our adversaries. Although the differences between the nature of the plant to be controlled by a traditional closed-loop controller and that to be controlled by a military command and control system are significant, there are obvious advantages in an ability to plan, execute, and replan many times faster than one's enemy.

The work reported here is one of the first instances where the benefits of higher rate loop closure have been quantified for a complex enterprise command and control application such as coordinating air attack operations, spanning the air operations enterprise from JFACC level to the strike package level. Our experimental results show that the benefits are substantial, and that they accrue even in the face of the types of model discrepancies that are to be expected in such applications. We should note that the results reported here assume perfect state estimation and feedback for own forces as well as for BDA.

9.1 Summary

In summary we found that:

- **Effective closed-loop solutions to realistic large-scale air operations planning problems can be largely automated.**
  - Automation allows decreasing controller cycle time from 24 h to 4 h.
  - Automation reduces the time to achieve Commander's Intent.
  - Closed-loop automation reduces sensitivity to unanticipated events.
- **Problem decomposition results in substantial reduction in computation time with little loss in performance.**
- **A controller can be designed to properly incorporate:**
  - Commander's Intent and user-definable levels of risk aversion.
  - Plan stability across successive planning cycles.
- **Optimal formulations for this class of problems are difficult to tractably solve, but can be used to help develop and evaluate heuristic algorithms that perform well.**

9.2 Suggestions for Future Work

We recognize that considerable additional effort is required to successfully transition Draper's JFACC closed-loop controller/planner to support military air operations. We have identified the following preliminary list of the technology enhancements and the implementation and risk reduction issues associated with transition to a fielded, operational system. Although it is technically feasible to address each of these, a significant undertaking is required to do so. Thus, our recommendation is that future efforts be directed toward the following:

Technology enhancements:

1. Strike CAP planning: allocating and scheduling strike CAP missions to better address time-critical targets.
2. Sharing of escorts across packages: zone-escort constructs and time sharing escorts.

3. Standoff-weapon release point planning.
4. Integration of cruise missile missions.
5. Multitarget bomber missions: devising bomber missions (i.e., for aircraft that have many weapons) and further enhancing the assembly point constructs we have currently in place.
6. Expected target value based on expected success of prosecuting the target: incorporating mission reliability estimates into the planning framework, including common cause considerations and responses such as consideration of more robust packages.
7. Tanker, counter-air CAP, and ISR planning: allocating and scheduling counter-air CAP, ISR, and tanker missions.
8. Plant modeling, identification and adaptation: exploration of stability and effectiveness of tactical adaptations (e.g., adapting threat models, thresholds, etc., with mission feedback).
9. Development of better plan monitoring visualization.
10. Inclusion of intelligent adversary.

Implementation and risk reduction:

1. Human-system interface: provide operators with insight into the planning process by allowing operators to change cost and risk parameters that affect the outcome of generated plans, modify plans and receive feedback regarding the expected performance of modified plans, and change thresholds for replanning in the execution monitoring and diagnosis functions.
2. Integrate with systems that: maintain the common operational/tactical picture, provide other C^4I and weaponeering information, and that support detailed route planning.
3. Interfaces to databases and communications systems.
4. Participation in military demonstrations and exercises.

10 References

[1] Lasdon, L. S., *Optimization Theory for Large Systems*, Macmillan, New York, New York, 1970.

[2] Tamura, H. and T. Yoshikawa, eds., *Large-Scale Systems Control and Decision-Making*, Marcel Dekker, New York, New York, 1990.

[3] Hall, William, D. and Adams, Milton B., "Closed-loop, Hierarchical Control of Military Air Operations," *Proceedings of Symposium on Advances in Enterprise Control,* San Diego California, November 14-15, 1999

[4] Adams, M.B., et al, "Closed-Loop Operation of Large-Scale Enterprises: Application of a Decomposition Approach," *Proceedings of 2nd Symposium on Advances in Enterprise Control,* Minneapolis, MN, July 10-11, 2000.

Appendix A: List of Acronyms

ADOC	Air Defense Operations Center
AOEM	Air Operations Enterprise Model
AWACS	Airborne Warning and Control System
BB&N	Bolt, Beranek & Newman
BDA	Battle Damage Assessment
BE	Basic Encyclopedia
C^2	Command and Control
C^3	Command, Control, Communications
CAP	Combat Air Patrol
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CVBG	Aircraft Carrier Battle Group
HARM	High-Speed Antiradiation Missile
HLA	High-Level Architecture
IAD	Integrated Air Defense
IP	Integer Program
ISR	Intelligence, Surveillance, Reconnaissance
JFACC	Joint Force Air Component Commander
KKT	Karush-Kuhn-Tucker
LOC	Lines of Communication
LOF	Local Output File
LP	Linear Programming
MIP	Mixed-Integer Programming
NPE	Number of Potential Engagements
POL	Petroleum, Oil, and Lubricants
SAM	Surface-to-Air Missile
SCL	Standard Conventional Loadout
SEAD	Suppression of Enemy Air Defenses
SFIP	Shared File Interprocess Concept
SOC	Sector Operations Center
TST	Time-Sensitive Target
UAV	Unmanned Aerial Vehicle
UCAV	Unmanned Combat Air Vehicle
XML	Extensible Markup Language